02－R 001

エキスパートシステム評価ガイドラインの研究開発報告書

平成 3 年 3 月

JIPDEC

財団法人 日本情報処理開発協会

この報告書は、日本自転車振興会から競輪収益の一部である機械工業振興資金の補助を受けて平成２年度に実施した「エキスパートシステム評価ガイドラインの研究開発」の成果の一部をまとめたものであります。

序

エキスパートシステムは、知識ベースと推論機構により問題解決を図るシステムであり、情報処理の知的化を実現するものとして大きな期待が寄せられています。エキスパートシステムは、処理アルゴリズムが存在しない、あるいは明確でない問題や、条件の組合せの数が膨大で逐一判断処理が行いがたい問題などに適しています。即ち従来技術の不得意な機能を補完し、結果として各アプリケーションの質の向上を実現することができます。

しかしながら、知識ベースの正当性や、それを用いての推論処理はブラックボックス性が強く、処理結果に対する信憑性を充分に保証することがきわめて難しいとされております。とくに、今後原子力や航空管制など非常に厳しい条件のもとに利用される場合、従来のソフトウェアの信頼性問題に加え、エキスパートシステムの推論結果の評価を如何に行うかは、緊急を要する検討課題であります。

このような背景のもとに、当協会ではＡＩ技術の視点を中心に、エキスパートシステムの開発者および利用者の双方の立場から、エキスパートシステム評価のためのガイドラインを提案することを目的として、昭和63年度から３年計画で、調査研究を実施してまいりました。

本プロジェクトの実施に当たっては、「エキスパートンステム評価ガイドライン研究委員会」を設置し、内外の資料分析や専門家との情報交換、ＩＪＣＡＩ等の国際会議での発表、委員会での討論等を重ねてまいりました。

ここに本調査研究の成果を報告書にとりまとめました。本ガイトラインが、今後のＡＩ技術やＡＩ利用の一層の普及発展に貢献し得ることを念願致します。最後に、本プロジェクトに格別のご尽力を頂いた研究委員会の委員各位に対して深甚なる謝意を表するものであります。

平成３年３月

財団法人　日本情報処理開発協会

「エキスパートシステム評価ガイドライン」の研究開発概要

知識工学の概念が１９７７年に提案されてから今日まで数多くのエキスパートシステムが開発されてきている。しかし、その一方では、充分な問題解決能力を持ちながらも、フィールドテストを経て、実用化の段階に達したエキスパートシステムの数は意外に少ない。この理由としては、さまざまなものが考えられるが、そのひとつは、開発中、あるいは、完成したシステムを客観的な基準で評価するための方法・基準が確立されていないことにある。特に、エキスパートンステム開発方法論と組合せた評価方法の整備が急務である。

このような現状認識のもとに、（財）日本情報処理開発協会では、１９８８年度から、エキスパートシステム評価のフレームワークの確立を目指して、調査研究を行なってきた。この間の活動の最終報告として、本報告書では、大規模なエキスパートシステム開発・運用のライフサイクルを通じて適用可能な、エキスパートシステム評価ガイトラインを提案する。本ガイトラインは、３４０項目におよぶチェックリストを中心としたもので、これには、エキスパートシステム開発・運用時に評価すべき重要な項目がすへて網羅されている。本報告書では、このガイドラインの基礎となる諸概念について知識工学的視点・ソフトウェア工学的視点から解説を行うとともに、チェックリストの利用方法について説明する。さらに、具体的なエキスパートンステム構築例に基づいたチェックリスト自身の評価を実施する。

本報告書の構成は以下のとおりである。１章では、総論として本調査研究活動の概要をとりまとめる。２章ではガイトライン作成の基本的な方針を述へる。この中で、本ガイトラインの対象とするエキスパートンステムの性質、エキスパートシステム評価に対する基本的な考え方、知識獲得活動とエキスパートンステム評価、エキスパートシステムの要求仕様とライフサイクル・トキュメントのあり方、本ガイドラインの利用方法に関する議論を行う。３章が本報告書の中心であり、エキスパートシステムの評価項目について詳細な解説を行う。そして、各評価項目が、ライフサイクル・評価者、ドキュメント、システム利用形態、テスト手法、問題のタイプの６つの観点からどのように分類・利用できるかを分析する。４章では、各委員が実際の開発にたずさわったエキスパートシステム事例を対象に、本ガイトラインの適用・評価結果について述へる。５章では、結論を述へる。付録Ａには、本年度実施した海外調査報告の結果をとりまとめ、付録Ｂには、エキスパートシステム評価に関する主要な文献の抄録を掲載する。付録Ｃには、エキスパートシステム評価の文献リストを掲載する。

人工知能技術の理論的研究の進歩は著しく速い。一般のソフトウェアの評価技術も確立していない現状において、現時点で、本報告書のようなソフトウェア工学的な立場から評価ガイドラインを作成するのは時期早尚であるとの意見もあろう。ところが、一方では、最近、産業界においては、計算機メーカ・大規模ユーザを中心に、エキスパートシステム開発方法論の確立を目指した事例研究が活発になされており、その中で、エキスパートシステム評価の問題が取上げられている。また、学界においては、エキスパートンステム開発の最大のボトルネックである知識獲得問題とエキスパートンステム評価の概念を結びつけるための研究がなされつつある。したがって、この２年半の間に、我国第一線の技術者・研究者が集まり、エキスパートシステム評価に関する大規模な調査研究を実施できたこ

と、また、不十分とはいえ本報告書に述べるようなガイトラインを作成・提案てきたことは、きわめて意義が深いと考えている。本報告書の成果が、今後の知識システム技術の実践・研究の場で利用されることを願ってやまない。

最後に、このような大きなテーマのもとで本調査研究を取りまとめる上で、2年半にわたって終始熱心に御討論いただいた委員の各氏、ならびに、各委員とともに御努力いただいた事務局の皆様に感謝の意を表する次第である。

１９９１年３月

エキスパートシステム評価ガイトライン研究委員会
委員長　寺野　隆雄

エキスパートシステム評価ガイドライン研究委員会名簿

（順不同、敬称略）

委員長	寺野　隆雄	筑波大学大学院　経営システム科学専攻講師
委　員	安藤　　明	（財）石油産業活性化センターＡＩ研究室研究員
	石垣　良信	日本アイ・ビー・エム㈱　システム・センター ＡＩＸ．クライアント／サーバー・サポート担当
	宇佐見仁英	富士通㈱　システム本部システム生産技術センター ＳＥ情報化推進部ＡＩ開発課課長
	荻野　　徹	三菱電機㈱　情報電子研究所システム部ＡＩ応用グループ主幹
	小林　慎一	㈱三菱総合研究所　情報科学部人工知能開発室室長
	末田　直道	㈱東芝　システム・ソフトウェア技術研究所 研究第一部ＡＩ第２担当主任研究員
	桃井　茂晴	日本電信電話㈱　ＮＴＴ情報通信処理研究所 知識処理研究部主幹研究員
	山口　高平	静岡大学　工学部情報知識工学科助教授
	山崎　知彦	㈱豊田中央研究所　研究７部７４３Ｇｸﾞﾙｰﾌﾟ ﾘｰﾀﾞ
	吉浦　　裕	㈱日立製作所　システム開発研究所第５部研究員
	渡辺　正信	日本電気㈱　Ｃ＆Ｃシステム研究所システム基礎研究部研究課長
	山本　欣子	（財）日本情報処理開発協会常務理事
	鍛治　勝三	（財）日本情報処理開発協会開発研究室主任研究員
事務局	金剛寺英雄	（財）日本情報処理開発協会開発研究室専任調査役

目　　次

1. 総　　論　1
1.1 はじめに　1
1 2 実運用型エキスパートシステムとその性質　3
1.3 エキスパートシステム開発のライフサイクルモデル　4
1.4 知識獲得活動とエキスパートシステム評価　8
1.5 ガイドラインの構成と利用法　10
1.6 エキスパートシステム評価のためのチェックリスト　13
1.7 エキスパートシステム評価ガイドラインの適用と評価　13

2. ガイドライン作成の基本的な方針　15
2.1 対象とするエキスパートシステム　16
2.2 評価に対する基本的な考え方　18
2.2.1 評価とは　18
2.2.2 ＶＶ＆Ｔとは　19
2 2.3 Ｖ＆Ｖと信頼性、品質との関係　21
2.2.4 評価の難しさ　22
2.2.5 知識獲得活動とＶ＆Ｖ　23
2.3 評価体系の基本的な考え方　26
2.3.1 評価対象　26
2.3.2 評価項目　28
2 3 3 点検項目／測定項目　30
2.3 4 テスト　31
2.4 要求仕様に対する基本的な考え方　38
2 4.1 要求仕様は作成できるか　38
2 4 2 エキスパートシステムの要求仕様の特徴　39
2.5 ライフサイクルの基本的な考え方　40
2.5 1 なぜプロトタイピングアプローチか　40
2.5.2 エキスパートシステム開発のライフサイクル　41
2.5.3 開発フェーズと評価対象との関係　48
2.5 4 評価者について　50
2 6 トキュメント作成に対する基本的な考え方　51
2.6.1 トキュメントは何処まで必要か　51
2.6 2 エキスパートンステム開発／保守用トキュメントの仕様　53
2 7 ガイトラインの使い方　60
2 7 1 ガイトラインの想定利用者と利用時期について　60
2.7.2 本ガイトラインの利用の仕方　62
2 7.3 本ガイトラインを更に有効に利活用するための方法　64

3. エキスパートンステムの評価項目　71
3.1 評価項目とその解説　72
3 1 1 知識ベースの妥当性評価　76
3 1 2 問題解決・推論方式の妥当性評価　91
3.1 3 結果の妥当性評価　100
3 1 4 要求仕様に対する達成度評価　109

3.1.5 知識ベースの保守性評価 …… 115
3.1.6 ヒューマンインタフェースの妥当性評価 …… 124
3.1.7 システムの総合評価 …… 128
3.2 ライフサイクル・評価者別評価項目 …… 147
3.2.1 ライフサイクル別評価のポイント …… 147
3.2.2 評価者別評価のポイント 154
3.3 ドキュメント別評価項目 …… 165
3.3.1 ライフサイクル・ドキュメント別による重点評価項目 …… 165
3.3.2 点検項目・測定項目 …… 168
3.4 利用形態別評価項目 …… 178
3.4.1 利用形態の分類 …… 178
3.4.2 利用形態別評価項目 …… 179
3.5 問題のタイプ別評価項目 …… 186
3.5.1 設計型システム …… 190
3.5.1.1 ドメインの特徴 …… 190
3.5.1.2 タスクの特徴 …… 191
3.5.1.3 設計型システムのV&V …… 192
3.5.2 計画型システム …… 200
3.5.2.1 ドメインの特徴 …… 200
3.5.2.2 タスクの特徴 …… 200
3.5.2.3 計画型システムのV&V …… 201
3.5.3 解釈型システム …… 208
3.5.3.1 ドメインの特徴 …… 208
3.5.3.2 タスクの特徴 …… 209
3.5.3.3 解釈型システムのV&V …… 210
3.5.4 診断型システム …… 216
3.5.4.1 ドメインの特徴 …… 216
3.5.4.2 タスクの特徴 …… 217
3.5.4.3 診断型システムのV&V …… 218
3.5.5 制御型システム …… 226
3.5.5.1 制御型システムの概要 …… 226
3.5.5.2 制御型システムの特徴 …… 226
3.5.5.3 制御型システムの基本タスク …… 228
3.5.5.4 制御型システムのV&V …… 229
3.5.6 複合型システム …… 237
3.5.6.1 ドメインの特徴 …… 237
3.5.6.2 タスクの特徴 …… 238
3.5.6.3 複合型システムのV&V …… 240
3.6 評価に用いられるテスト方法 …… 246
3.6.1 概要 …… 246
3.6.2 エキスパートシステムのテスト方法の分類体系・次元 …… 246
3.6.3 妥当性を検査するテスト方法 …… 248
3.6.4 静的テスト／分析および動的テストの概要 …… 250
3.6.5 静的テスト／分析 …… 251
3.6.6 動的テスト …… 255
3.6.7 評価項目に対する重点テスト方法 …… 265

4. 本ガイドラインの適用と評価 271
4 1 ジョブショップスケジューリングシステムへの適用 272
4 1 1 プロフィール 272
4.1.2 ガイトラインの適用結果 274
4 2 鉄鋼プラント異常診断エキスパートシステムへの適用 281
4.2 1 プロフィール 281
4 2 2 ガイトラインの適用結果 286
4 3 LSI論理合成エキスパートンステムへの適用 294
4 3 1 概要 294
4.3 2 LSI論理合成エキスパートシステム(EXLOG)とは 294
4.3 3 本ガイドラインによるEXLOGプロトタイプの評価 296
4 3.4 EXLOG実用版への発展 298
4 4 水力鋼構造物評価診断ンステムへの適用 306
4 4.1 プロフィール 306
4.4 2 ガイトラインの適用結果 311
4.5 ガイドラインの評価 314

5. 結論 315

付録A. 海外調査報告 317
1. 調査概要 317
2. 本ガイトラインに対する意見交換 319
2.1 ブリティシュコロンビア大学 319
2 2 シムフレックステクノレッジ社 320
2 3 南カルフォルニア大学 321
2 4 ストラクチャートンステム&ソフトウェア社 324
3. MES'90国際会議 325

付録B. エキスパートンステム評価の文献調査 340
1. エキスパートシステムの失敗の原因 341
2. 原子力発電所向けエキスパートンステム開発におけるV&V方法論 345
3. 知識ベースンステムのV&Vアプローチ 352
4. エキスパートシステムの検証と妥当性検査 358
5. 専門家間の概念構造の比較 362
6. 決定理論を用いた経験的知識の正当性の評価法 365
7. エキスパートンステムの妥当性試験手法 370
8. エキスパートシステムの検証及び妥当性検査 376
9. 知識ベースのテスト 381
10. エキスパートンステムの性能に対する妥当性検査 387
11. KB-Reducer 知識ﾍﾞｰｽの非一貫性及び冗長性の検査に対する新ｱﾌﾟﾛｰﾁ 390
12. Digital社におけるコンフィギュレーションエキスパートンステム 395
13. エキスパートンステムのテスト、検証、引渡し-Mentor履歴の事例- 401
14. ORにおける視覚対話型モデリング手法: その成功と可能性 405

付録C. 文献一覧表 409

1　総　　論

1. 総　論

1.1 はじめに

知識工学が提唱されてから今日まで、数多くのエキスパートシステムが開発されてきた。これらを通じて、人工知能技術は、従来では、取組む気にもならなかったような問題にも、専門家－知識技術者というペアで挑戦できるような新しいパラダイムを提供してきたといえる[ICOT-JIPDEC 1989]、[ＡＩ白書 1989]。
また、エキスパートシステムの開発に直接的にかかわってきた人々の間では、人工知能の科学としての成功がようやく実用システムとしての成功に結びついてきたこと、ならびに、人工知能技術を指向したシステムに対してニーズ指向型のシステムが急増していることが認識されるようになった。また、最近では、ルール・フレームなどの典型的な人工知能手法を整理し、開発方法論としてシステム開発の現場で利用できるように整備していく動きが、ツール提供各社や大規模ユーザの間で活発になっている。

しかし、その一方では、充分な問題解決能力を持ちながらも、フィールドテストを経て、実用化の段階に達したエキスパートシステムの数は意外に少ない。この理由としては、さまざまなものが考えられるが、そのひとつは、開発中、あるいは、完成したシステムを客観的な基準で評価するための方法・基準が確立されていないことにある。特に、エキスパートシステム開発方法論と組合せた評価方法の整備が急務であると考えられる。

このような現状認識のもとに、（財）日本情報処理開発協会では、１９８８年度から、エキスパートシステム評価のフレームワークの確立を目指して、調査研究を行なってきた。この経緯は、図1-1 のようにまとめられる。我々の調査研究の目的は、「できるだけ広い範囲のエキスパートンステムの開発・運用時に、評価規準として使用できる一般的なガイドラインを開発すること」であり、そのために以下のような活動方針をとった。

（１）エキスパートシステム評価技術の現状を分析し、知識工学的立場・ソフトウェア工学的立場から、その統合化を指向する。
（２）エキスパートシステム評価手法の事例分析を実施する。
（３）エキスパートシステム評価のガイトラインを提案する。
　(3-1) エキスパートシステム評価のための包括的なチェックリストの作成
　(3-2) チェックリストの利用方法の提案
　(3-3) 評価ガイドライン自身の事例による評価

本報告書は、２年半におよぶ研究調査の最終報告である。本報告書では、上の方針にしたがって、大規模なエキスパートシステム開発・運用のライフサイクルを通じて適用可能な、エキスパートシステム評価ガイドラインを提案する。本ガイドラインは、３４０項目におよぶチェックリストを中心としたもので、これには、エキスパートシステム開発・運用時に評価すへき重要な項目がすへて網羅されている。本報告書では、このガイトラインの基礎となる諸概念について知識工学的視点・ソフトウェア工学的視点から解説を行うとともに、チェックリストの利用方法について説明する。さらに、具体的なエキスパートシステム構築例に基づいたチェックリスト自身の評価を実施する。

調査研究の方法：

（１９８６年－１９８９年：
ICOT-JIPDEC AIセンターにおける知的情報処理システム調査研究委員会にて、知識システムの開発方法論・知識システム構築ツールの比較分析・知識技術者育成ガイドラインの調査・研究ならびに代表的な知識システムの事例分析を実施。[Kobayashi 1990]）

１９８８年度：
エキスパートシステム評価ガイドラインに対するフィージビリティスタディの実施。[JIPDEC 1989]

１９８９年度：
- ＶＶ＆Ｔ理論・技術に対する現状分析・調査
- 代表的なエキスパートシステムに対する評価手法の事例調査・分析
- エキスパートシステム評価ガイドラインの中間報告
  [Terano 1989], [JIPDEC 1990], [Terano 1990]

１９９０年度：
- エキスパートシステム評価ガイドラインの提案
- チェックリストの作成と事例に基づく評価
- チェックリストに基づく評価手順の提案
- ＶＶ＆Ｔ理論・技術に対する現状分析・調査

図1-1　本研究委員会における調査研究活動のあらまし

一般にシステムを評価する場合の視点としては、社会的な視点、経済的な視点、技術的な視点の３つが存在する。これら３つの視点を総合しなければ、開発中あるいは、完成したシステムの評価を厳密な意味で実施することはできない。また、このような評価の問題はソフトウェア工学の大きなテーマにもなっている。

エキスパートシステムに限れば、社会的な視点としては、例えば、利用者が知識ベースの知識を基準として納得できるのか、信用がおけるのかなとの問題がある。経済的な視点としてはエキスパートシステムがどれたけの利益を生み出すのか、開発の費用効果をどう判断するのかなどの問題が生ずる。第３の人工知能技術から直接派生する技術的な諸問題を主な検討の対象とする。これは、技術的な視点からのエキスパートシステム評価手法が定まっていない現状においては、まず、これを評価ガイドラインとしてまとめることが重要と考えたからである。

技術的な視点に注目すると、エキスパートシステムの評価の問題は、特にシステムのVerification（検証）とValidation（妥当性検査）（以下Ｖ＆Ｖと呼ぶ）、ならびに、Ｖ＆Ｖを行う手段としてのテスト(Testing)が中心となる[Deutsch 1982]。これらを総称して本論文では「評価」という用語を用いることにする。

検証とは、システムの内的な正しさを保証する作業、すなわち、システム開発時に生ずるエラーをなくす作業を指し、開発工程において正しくシステムを構築するための確認作業である。

妥当性検査とは、完成したシステムが（陽には表現しにくい）ユーザの真の要

求の目的を満たしているかを確認する作業である。すなわち、妥当性検査では、システムは正確に要求が反映しているのか（正確性）、ユーザにとり有用なものであるといえるのか（有用性）が重要となる。そして、正確性、有用性を確認するためには、通常要求仕様の達成状況を評価することとなる。

しかし、エキスパートシステムを対象とする限りにおいて、我々は、ここでジレンマに陥らざるを得ない。エキスパートンステムを含む知識システム技術は、もともと悪構造・悪定義な対象に対するアプローチとして発展してきた手法であり、仕様が不明確であるとの前提に開発に踏切る場合が多いからである。

したがって、エキスパートシステム評価ガイトラインを提案しようとすれば、対象とするエキスパートシステムの性質・開発方法・ライフサイクルなとの諸概念を明確にする必要が生ずる。本委員会における研究調査活動の大部分はそのための作業に費やされた。その結果として、本報告では、Ｖ＆Ｖ技術、テスト手法、ライフサイクルモデル、知識獲得活動、エキスパートシステムの型なとさまざまな観点から利用が可能な、チェックリストベースのエキスパートシステム評価ガイトラインを提案することができた。以下の節では、その概要を取りまとめる。１.２ では、本ガイドラインの対象とするエキスパートシステムの性質について議論する。Ｖ＆Ｖを実施するためには、その開発工程において行われる作業の内容・入出力をあらかじめ決めねばならない。そこで、１.３ では、現在我国で通常実施されているシステム開発方法に即した、実践的なライフサイクルモデルを提案し、知識獲得活動とＶ＆Ｖとの関連について考察する。１ ４ ては、評価ガイドラインの構成と利用法について説明する。

１.２　実運用型エキスパートシステムとその性質

エキスパートシステムあるいは、知識ンステムとは、通常以下のようなンステムとして定義される。すなわち、”通常人間の知性を用いなければ解けないような特定分野の問題を解決するのに、専門家の知識をコンピュータに組み込んで、人間に代わってあるいは人間を支援して問題解決を行うことを目的とした計算機システムである”（たとえば、[Hayes-Roth 1983]）。このような定義のもとては、エキスパートシステムの評価は、専門家の知識との比較によってのみ成り立つものとなる。

それに対し、実環境において運用されるエキスパートシステムては、問題解決能力が重要となる。そこで、我々は、以下の性質を満足するようなコンピュータシステムを、実運用型エキスパートシステム(Operational ES)と呼ふことにする。

- 人工知能技術をシステム実現の技術的な基盤とすること。
- 利用者の問題解決に対する要求に対応して開発されるシステムであること。
- 利用者（専門家／非専門家）にとって、問題解決の支援または代替を行う能力を持つこと。

本報告書では実運用型エキスパートンステムを評価対象とする。実運用型エキスパートシステムは、ユーザの機能要件、性能要件等のニーズを充分満足させることを目標とする。したがって、たとえば性能要件がユーザに満足されないとすると作り直しを行なう必要も生ずる。このようなンステムに対しては、客観的な

評価の必要性が高く、本調査研究の対象としても適切である。

一方、エキスパートシステムにどのような機能をどのように持たせるかという観点でシステムの開発を行う、人工知能技術を重視したケースも存在する。これを研究開発型エキスパートシステム（Exploring ES）と呼ぶことにする。

実運用型エキスパートシステム型と研究開発型エキスパートンステムは、たとえ同じ機能を実現していても、それに至るライフサイクルや、要求仕様に対する考え方のアプローチには大きく異なる。本報告では、研究開発型エキスパートンステムは評価という概念にそぐわないので、その対象としない。

現状では、実運用型エキスパートシステムの多くは、知識ベースと推論機構とを備えた、いわゆる知識ベースシステムである。しかし、以下の議論であきらかになるように、ここで提案する評価ガイトラインは、知識ベースンステムに限らず、ファジィやニューラルネットを利用するシステムについても一部適用可能なものである。

また、一般に、エキスパートシステムは実用システムに近つけば近つくほど、従来型システムとの統合化が重要となる。それに対して、本報告で述べる評価ガイトラインは、議論の発散を避けるために、とりあえず、エキスパートンステムとして作られている部分のみを評価対象とする。従来型システムとして最初から作られている部分については、（確立した手法はないものの）従来型の評価手法が適用可能と考えられる。たたし、実運用システムでは、開発の最終段階において人工知能手法をすてて従来手法で再開発されるケースもある。このガイトラインでは、このような立場から最終的に作りなおされたシステムも実運用型エキスパートンステムと考えて、同じ評価手法を適用するものとする。

より詳しい議論については２　１　を参照されたい。

１　３　エキスパートシステム開発のライフサイクルモデル

エキスパートシステムの対象分野は問題の構造が不明確な、悪構造（Ill Structured）問題であり、同時に、問題自体を明確に規定することが難しい悪定義（Ill Defined）問題である。したがって、システム開発当初においては要求を明確に定義付けることが困難で、変更が頻繁に行われることとなる。一方、従来システムにおいては、要求仕様はそのシステムを評価する基準のベースになる。

このような現状に鑑み、我々は、プロトタイピングに基づくフェーズトアプローチによってライフサイクルモデルを構成することとした。この考えのもとでは、開発当初においてシステムとしての本質的な要求を定義し、システムを徐々に成長させていく各フェーズで、段階的に要求仕様を明確化していくアプローチを採用する。そして、各フェーズが終了するごとに、直前の段階で定義された要求仕様に対するＶ＆Ｖが実施され、次のフェーズへむけての仕様が決定する。

プロトタイピングの考え方は、従来型ソフトウェアの開発においても成立するものであるが（たとえば、[Birrel 1985],[Martin 1982],[Mayhew 1987]）、エキスパートシステムの場合には、特に、(1) 問題解決に必要な情報（知識）とそれに必要なアルゴリズムの性質が開発初期には明確になっていない点、(2) ンステムが成功をおさめるには、継続的な保守・拡張の作業が不可欠となる可能性が強い点の２つが特徴的である（[Weitzel 1989]も参照のこと）。

一般的にプロトタイプを作成する目的として三つの立場が考えられる。

- 技術的に未経験な部分、不安な部分を切り出して、その部分の簡単なモデルを作成して技術的可能性を検証する。
- システムの技術ポイントを抽出するために、既知の部分から作成し、不明な部分はブラックボックス化しシステムイメージを構築する。
- システムのイメージを共有化しシステム運用時に仕様上のトラブルを防止するために、システムの動作イメージが見えるようなプロトタイプを作成する。

Ill Structured な問題に対応するという観点からは上の３つのうち最初の立場で開発する場合が主流となる。また、研究的要素が強いエキスパートシステムについては、ライフサイクルという概念をあてはめにくい面も存在する。これについては、初期的な要求仕様をまとめるまでの試行的な研究開発を、特に、探査型プロトタイプ（Exploratory prototype）と呼び、探査型プロトタイプは研究開発型エキスパートシステムと同様の性質をもつので、ここでの評価の対象とはしない。

エキスパートンステム開発のライフサイクルと関連するトキュメント・評価対象を図1-2 に示す。

開発フェーズ	区分	調査計画（探査型プロトタイピング）	プロトタイピング		実用ンステム開発		運用テスト	運用 洗練化
			実験型プロトタイピング	発展型プロトタイピング		改善 / 再開発		
評価対象	ソフトウェア	—	第Ⅰ版プロトタイプ	第Ⅱ版プロトタイプ	実用システム		実用ンステム	デリバリンステム
	ドキュメント	企画提案書（第0版要求仕様書）	第Ⅰ版要求仕様書	第Ⅱ版要求仕様書	実用版要求仕様書			変更要求仕様書
			第Ⅰ版知識分析書	第Ⅱ版知識分析書	実用版知識分析書			
		調査報告書	第Ⅰ版問題解決分析書 第Ⅰ版ンステム設計書 第Ⅰ版テスト文書	第Ⅱ版問題解決分析書 第Ⅱ版ンステム設計書 第Ⅱ版テスト文書	実用版問題解決分析書 実用版システム設計書 実用版テスト文書 運用マニュアル 保守マニュアル		運用テスト版テスト文書	運用実績書 保守報告書 改訂版ドキュメント一式

図 1-2 開発フェーズと成果物・評価対象

注）□は、必須トキュメントである。他は必要に応して使うことも可能である。

（１）調査・計画フェーズ

このフェーズは従来のシステム開発における初期フェーズと同しで、要求仕様、システムの位置づけ（他のシステムとの関連性）、開発プロジェクト体制、開発スケジュールを明らかにする。この段階で、企画提案書（第０版要求仕様書）が作成される。企画提案書は、企画グループが作成する。企画グループは、利用部門のマネージャ、開発部門のマネージャ、運用部門のマネージャ、ドメインの専門家、エンドユーザのメンバーで構成される。この段階で企画提案書を固めるために、探査型プロトタイピングが行われる場合もある。

（２）実験型プロトタイピングフェーズ

このフェーズでは、実験的にプロトタイプを作成する。作成されるプロトタイプは、目的を重点的に絞り、知識量、性能、マンマシン機能等の側面では実用レベルに達する必要はない。ここでは技術的に核になる知識・推論機能が実現され、技術的な実現可能性の検証に重点がおかれる。このプロトタイプにより、実用化のめどをたてる事が重要である。設定された目的が達成できるまて、実験型プロトタイプは繰返し作成される。

（３）発展型プロトタイピングフェーズ

このフェーズでは、規模的には小規模でも、最終システムの機能を一応実現したレベルのプロトタイプを作成する。このフェーズのおわりには、プロトタイプを関係者に対しデモを行い、実用システムの開発の可否が判断できるレヘルに達しているものとする。また、エントユーザ、開発者、専門家のそれぞれが、このフェーズ後半にはシステムイメージが明確になり実用ンステムの要求仕様を詳細に記述できるレベルになるよう、プロトタイプの作成を行う。なお、このフェーズの途中で、必要の都度デザインレビューが行われる。

（４）実用システム開発フェーズ

プロトタイプが評価用またはデモ用であるのに対して、実用ンステムとは実際に業務に組み込まれて利用されるシステムであると定義する。このフェースの作業は従来のシステム開発と基本的に変わりはない。ただ、このフェースにいたるまでにプロトタイプを作成しているため、その結果（仕様・知識・技術・プロクラム等）を有効に利用することができる。ただし、必ずしもプロトタイプンステムを改良して実用システム化するわけではない。実用システムの仕様に合わせて、言語・ツール、推論機構等を見直して、再構築するアプローチをとる場合もある。

（５）運用テストフェーズ

このフェーズでは、システムを現場でテストする。テストデータによる検証、従来システムとの並行実行によって、システムを評価し運用に耐えられるレヘルまで洗練化する。テストは長期間にわたる場合が多い。

（６）運用・洗練化フェーズ

このフェーズでは、システムを実運用しながら、システムの問題解決能力を一層向上させていく。システムは開発グループから運用グループに引き渡され、必要な保守作業・能力向上作業はユーザと運用担当者グループの協力て行われる。システム変更に対する最終責任者は運用担当マネージャである。

これらについて、詳しくは、２４～２６を参照されたい。

１．４　知識獲得活動とエキスパートシステム評価

エキスパートシステム開発には、知識、すなわち、問題解決に必要な情報を、専門家などの知識源から収集・定式化し、それを人工知能技術が提供する手法で取扱える形に変換し、さらにコンピュータで扱えるように構造化する作業が必要である。また、エキスパートシステムは一度動きだしてからも、知識の誤りや矛盾、あるいは、知識の増大に対応して、常に、変更され続けるのが普通である。知識獲得活動は、知識の収集・知識の変換・知識の構造化・知識の洗練化の４段階に分けることができる[Kobayashi 1990]。

知識獲得活動とエキスパートシステム開発のライフサイクル、ならびに、Ｖ＆Ｖ作業との関連は図1-3 のように表現できる。すなわち、エキスパートシステム開発作業が上流から下流に進むにつれて知識の収集・知識の変換・知識の構造化・知識の洗練化の順に中心となる作業が変化していく。各段階で、獲得された知識は、それぞれ適切な方法で検査・確認されなくてはならない。この意味で、知識獲得活動に関する議論はすべて、Ｖ＆Ｖと深く関連する。

現在のところ広い分野・範囲で利用できる知識獲得支援ツールは非常に少なく、エキスパートシステムのＶ＆Ｖを陽にサポートしているものは存在しない。そのため、以下で述べる評価ガイドラインでは、Ｖ＆Ｖと知識獲得支援ツールとの関連を明確にせず、かわりに、さまざまなテスト手法を分類して、評価項目と関連づけた。これらについては、詳しくは２２．５、２３４ならびに３５を参照されたい。

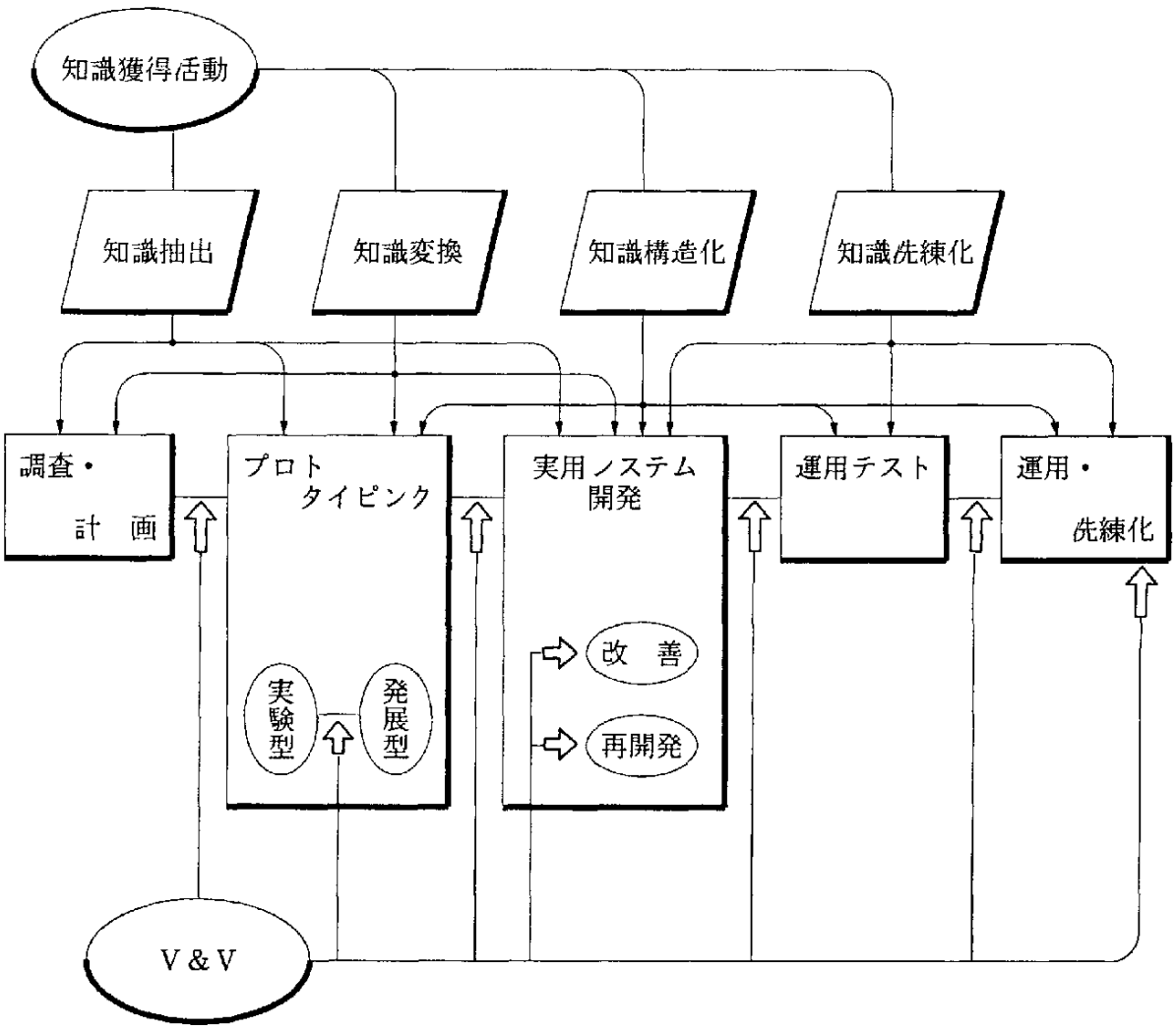


図１－３　知識獲得活動と開発フェーズ・V&Vの関係

1 5　ガイドラインの構成と利用法

我々の設定した評価体系の構成を図1-4 に示す。この内容に関する詳しい議論は3章においてなされる。

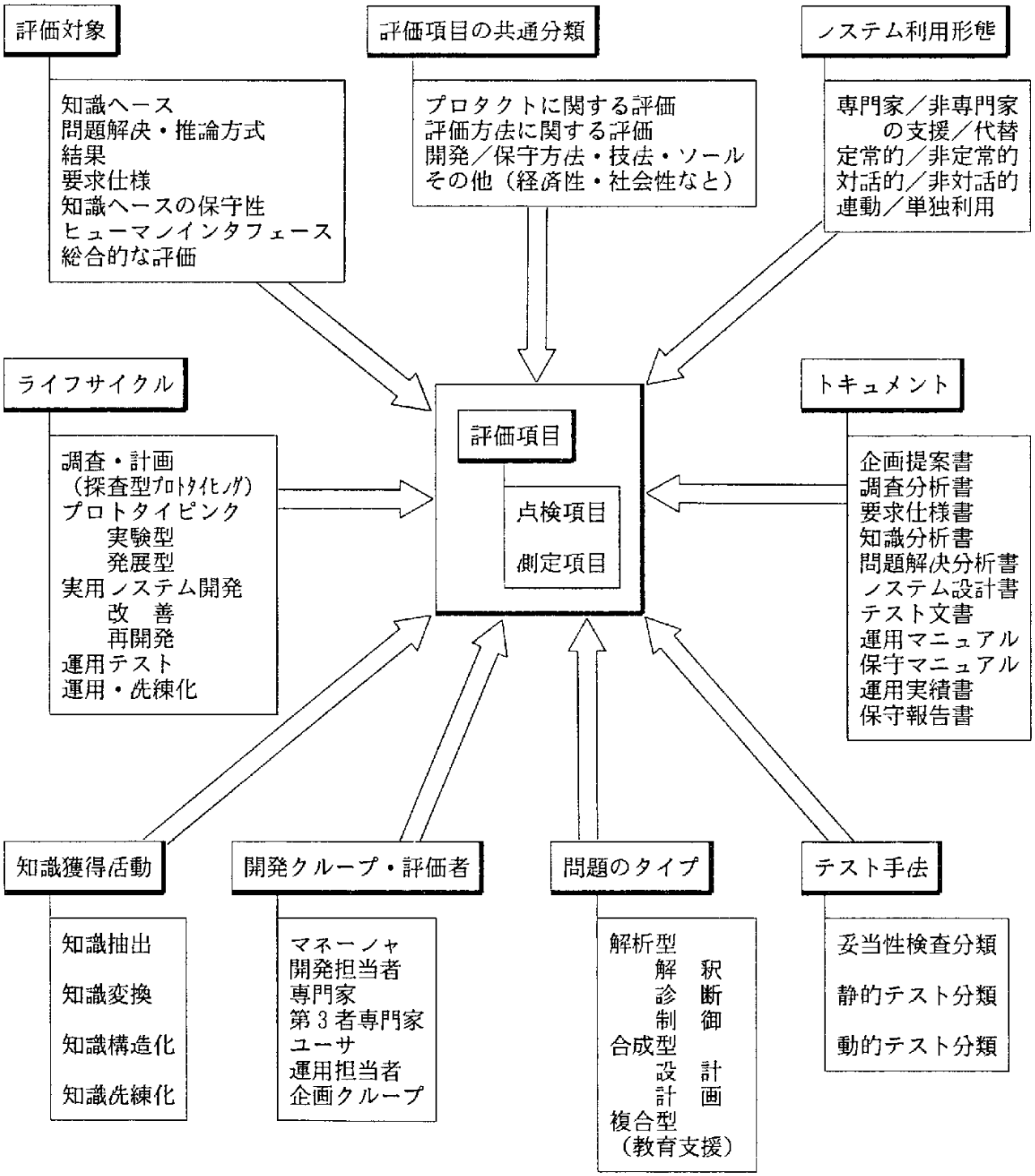


図1－4　評価カイトラインの構成

この設定にあたっては「品質機能展開」の手法[IPA 1989]を用いた。すなわち、品質を保証する対象（評価対象）を定め、評価対象をブレークダウンして、評価項目を設定し、更に点検項目、測定項目に展開した。この結果、階層構造化され、細分化された項目を「評価項目の体系」と呼ぶ。この項目の評価を実施する手段が、テストに相当する。ここでいうテストとは、コンピュータを用いての作業のみならず、人手によるレビューも含める。

図1-4 のうち、ライフサイクル、知識獲得活動についてはすでに考察したので、以下では残りの部分について説明を行う。

（１）評価対象

我々は、主に技術的な観点からエキスパートシステム自体を評価することとしているが、その場合の評価の対象は、基本的には計算機システムの考え方により、(1)システム要求に関連する部分、(2)その結果、及びその処理の仕方（これをエキスパートシステムの枠組みにあわせて、(3)知識ベース、(4)問題解決・推論機構、(5)マンマシンインタフェースに分類する）、(6)保守の問題、及び(7)システム全体に係わる問題に大きく７つに分類する。７つの評価対象をさらにブレークダウンした項目を「評価項目」と呼ぶ。

（２）評価項目

各評価項目は、以下のようにして、共通の４種類に分類できる。
まず、プロダクトに関する評価（結果系、Ｐ）と要因に関する評価（要因系）とその他経済性・社会性に関する評価（Ｏ）の３つに大きく分類される。要因系は、さらに、評価方法そのものを評価する場合（Ｅ）と方法論・技法・ツールの評価の場合（Ｍ）とに分けられる。評価項目を分類・整理した結果を図1-5 に示す。

各評価項目はさらに、点検項目と測定項目という次元で分類される。
点検項目とは、システムを構築する前のデザインレビューの際、または構築後評価をする際に、定性的に確認チェックするときに用いる。測定項目とは、知識ベース規模や推論速度など、実際に定量的に測定が出来る評価項目のことてある。

（３）テスト

エキスパートシステムのテスト手法にはさまざまなものが存在するが、本報告書では、妥当性を検査する方法による分類７種（内容的妥当性テスト、基準に基づく妥当性テスト、構成概念的妥当性テスト、表面的妥当性テスト、未来予測的妥当性テスト、併存的妥当性テスト、過去予測的妥当性テスト）と、テスト対象を実行するか否かによるテスト方法の分類（静的テスト／分析と動的テスト）とに分類し、代表的な手法を網羅的に述へる。

なお、静的テストとはテスト対象を実行させないでテストすることてあり、人手によるレビュー／ウォークスルーをする場合と、静的テスト用ツールを使用する場合がある。また動的テストとは、テスト対象を実行させてテストすることであり、代表的なテストとして、ブラックボックステスト／機能テスト、ホワイトボックステスト／構造テストなどがある。

本報告書で紹介するテスト手法については、必ずしも、すへてのテスト方法を任意に組合せて実行できるわけではない。対象とする問題の性質に応して、適切な組合せを選択する必要がある。

（４）システムの利用形態

エキスパートシステムに対する要求はその利用形態によって異なる。そこで、専門家／非専門家の作業の支援／代替、システムを定常的に利用するか否か、対話的に利用するか非対話的な利用か、他システムと連動して利用されるか単独システムとして利用されるかという、４つの次元から、システム利用形態の分析を行った。

（５）開発グループ・評価者

エキスパートシステムの開発グループのメンバー・評価者を以下のように分類する。

① マネージャ　：システム開発に責任をもち、次のフェースへ進むか否か決断を行う。
② 開発担当者　：知識技術者。主としてデザインレビューや、ウォークスルーを行う。
③ 専門家　　：対象システムの知識提供者である。
④ 第３者専門家：当該分野の専門家ではあるが知識提供者とは異なる。
⑤ ユーザ　　：専門家支援システムでは、ユーザは専門家で、専門家の代行システムでは、ユーザは非専門家である。
(6)運用担当者　：実用システムの管理・運用を行い、システムの変更の責任を担う。
(7)企画グループ：企画グループは、システム開発に先立ち、システムの調査・計画を行うグループであり、下記のメンバーから構成される。
利用部門のマネージャ、開発部門のマネージャ、運用部門のマネージャ、エンドユーザ、専門家。

（６）問題のタイプ

エキスパートシステムの適用範囲は広いので、そのタイプの違いによって評価手法が異なることは、十分に考えられる。そこで、本ガイドラインは、[Kobayashi 1990]など、さまざまな文献で提案されている標準的な分類方法にしたがって、エキスパートシステムのタイプを分類し、各タイプごとに重要な評価項目について考察を行った。

具体的な分類としては、分析型の問題として、解釈・制御・診断問題の３つを、合成型の問題としては、設計・計画問題の２つを用いた。さらに、これらに含むことが難しい複合型問題として、特に、教育支援システムを中心に分析を行っている。

（７）ガイドラインの利用方法について

本ガイドラインの利用者は、エキスパートシステムの開発にたずさわる開発担当者もしくはプロジェクト管理担当者を想定した。このような担当者レベルの技術者・管理者こそ、エキスパートシステムのＶ＆Ｖの概念を明確に理解することが重要である。また、ガイドラインを利用することによって、より品質の高いシステムを実現する義務をおっている。
さらに、この想定利用者は、（必ずしもエキスパートシステムに限らない）ソフトウェア開発プロジェクトの経験を持ち、かつ、人工知能の基礎概念についてひととおりの知識を持つことを仮定している。

また、ガイドラインの利用時期としては、システムの調査・計画が終了した時点、もしくは、対象システムに対する探査型プロトタイピングの作業が終了した時点以降を想定している。そして、ライフサイクルにしたがって、実運用システムが完成し、運用・保守にはいるまで、各フェーズごとに利用できるものを目指した。

ここで提案した評価項目は大規模かつ包括的なものなので、これに含まれる項目の全てについて、厳密な評価を実施することは、時間的にもコスト的にも現実的でない。したがって、利用にあたっては、特に効果が高いと思われる視点を予め検討しておく必要がある。評価項目の抽出に当たっては、図1-4 に示したように、利用目的に応じた、さまざまな分類体系を準備している。推薦できるのは以下のような方法である。(1) 特に着目したい視点を選択し、(2) 対応する評価項目を抽出する。さらに(3) 関連の深い点検項目及び測定項目について評価を行う。

利用方法に関する詳しい議論は２．７ において行われる。

１ ６　エキスパートシステム評価のためのチェックリスト

本チェックリストにおける評価項目は、合計、３４０項目にのぼり、各項目ごとに、テスト手法との関連性を明確にしている。点検項目については、種々の文献で提案されている項目をすべて含むものとなっている。測定項目については、具体的な測定手法そのものが、現在の最新の研究テーマとなっているものが多く、抽象的な記述にとどまっている項目も存在する。重点評価項目については、トキュメント別、システム利用形態別、システムのタイプ別、テスト手法別のものが準備されている。

この内容については、３ １において詳細に論ずる。

１ ７　エキスパートシステム評価ガイトラインの適用と評価

３章で提案するエキスパートシステム評価ガイトラインの有用性を評価するために、本研究委員会委員が実際に研究開発にたずさわったエキスパートシステムを事例として、本ガイトラインを適用することとした。この結果については、４章で詳細に検討されている。ここで対象としたエキスパートシステムは、合成型問題として、ジョブショップスケジューリング問題とＬＳＩ論理合成問題の２つ、解析型問題として、鉄鋼プラント異常診断問題と水力鋼構造物評価診断問題の２つ計４つを取上げる。これらのシステムはいずれも実運用型エキスパートシステムとして開発されたものである。しかし、これらは、すでに完成したシステムであるので、ガイドラインを開発中に適用することはできなかった。そこで、ガイドラインの分析に当っては、開発途中のエキスパートシステムに適用することを想定して、どのような効果・欠点が考えられるかを十分に考慮することとした。

《参考文献》

[Birrel 1985] Birrel,N.D., Ould, M.A.: A Practical Handbook for Software Development. Cambridge University Press, 1985.

[Deutsch 1982] Deutsch, M.S.: Software Verification and Validation - Realistic Project Approaches. Prentice-Hall, 1982.

[Hayes-Roth 1983] Hayes-Roth,F., Waterman, D.A., Lenat,B.(eds.):Building Expert Systems. Addison-Wesley, 1983.

[IPA 1989] 情報処理振興事業協会技術センター（編）：品質機能展開による高品質ソフトウェアの開発手法．コンピュータエージ社，1989.

[JIPDEC 1989] 日本情報処理開発協会：エキスパートシステム評価に関する研究開発，1989.

[JIPDEC 1990] 日本情報処理開発協会： エキスパートシステム評価ガイドラインの研究開発報告書．01-R003，1990.

[Kobayashi 1990] 小林重信，寺野隆雄（編）：知識システムハンドブック．オーム社，1990.

[Miller 1989] Miller,L.A.: A Comprehensive Approach to the Verification and Validation of Knowledge-Based Systems. Proc. IJCAI'89 Workshop on VV&T of KBSs, 1989.

[O'Keefe 1989] O'Keefe,R.M., O'Leary, D.E.: The Verification and Validation of Expert Systems. 11th IJCAI Tutorial, MP4, 1989.

[O'Leary,T. 1990] O'Leary,T. L., Goul, M., Moffitt, K.E., Radwan, A. E.: Validating Expert Systems. IEEE Expert, Vol.5, No.3, pp. 51-58, 1990.

[Terano 1989] Terano, T., Kobayashi, S.: Problem Analyses, Tool Evaluation, and Verification & Validation Study: Three Steps for Knowledge-Based Systems Development Methodology. IJCAI-89 Workshop on VV&T of Knowledge-Based Systems, 1989.

[Terano 1990] Terano, T., Kongoji, H., Kaji, K., Yamamoto, K.: Developing a Guideline for Expert System Evaluation - a Midterm Report. Proc. PRICAI-90, pp. 164-169,1990.

[Weitzel 1989] Weitzel,J.R., Kerschberg,L.: Developing Knowledge-Based Systems: Reorganizing the System Development Life Cycle. Comm. ACM, Vol.32, No. 4, pp. 482-488, 1989.

2　ガイトラインノ作成の基本的な方針

2.　ガイトライン作成の基本的な方針

第２章では、ガイドライン作成の基本的な方針を述へる。この中で、本ガイトラインの対象とするエキスパートシステムの性質、エキスパートシステム評価に対する基本的な考え方、知識獲得活動とエキスパートシステム評価(２２５節)、エキスパートシステムの要求仕様とライフサイクル・トキュメントのあり方、本ガイドラインの利用方法に関する議論を行う。

目次構成は以下の通りである。

2.1 対象とするエキスパートシステム
2.2 評価に対する基本的な考え方
2.3 評価体系の基本的な考え方
２４ 要求仕様に対する基本的な考え方
2.5 ライフサイクルの基本的な考え方
2.6 ドキュメント作成に対する基本的な考え方
２７ ガイドラインの使い方

2.1 対象とするエキスパートシステム

知識工学が提唱されてから今日まで、数多くのエキスパートシステムが開発されてきたが、エキスパートシステムあるいは、知識システムとは、通常以下のようなシステムとして定義される。すなわち、「通常人間の知性を用いなければ解けないような特定分野の問題を解決するため、専門家の知識をコンピュータに組み込んで、人間に代わってあるいは人間を支援して問題解決を行うことを目的とした計算機システムである。」（例えば、［Hayes-Roth 1987］）。このような定義のもとでは、エキスパートシステムの評価は、専門家の知識との比較によってのみ成り立つものとなる。

それに対し、最近では、エキスパートシステムが各企業の情報戦略として、または高度な情報処理システムとして開発が進められるようになった。この背景には、第２世代の汎用エキスパートシステムツールの登場、第３世代と言われるドメインシェルの登場により、従来に比較してより開発者側の負担が少なく、より知識中心に開発を進めることが可能となってきたことがある。すなわちこれまで多数のシステムが試行錯誤的に開発された経験を生かし、企業においては、情報処理システム部門のンステム開発者またはそれぞれの領域の専門家自身なと、元々ＡＩの専門家でない人達でも開発が可能となり、開発技法の整備と共に、人的、開発環境、設備投資などに過大なリスクを伴わず開発することが出来るようになってきた。このような実環境で運用されるシステムでは、問題解決能力が重要となる。そこで、我々は、以下の性質を満足するコンピュータンステムを実運用型エキスパートシステム（ＯＥＳ：Operational ES）と呼ぶことにする。

- 人工知能技術をシステム実現の技術的な基盤とすること。
- 利用者の問題解決に対する要求に対応して開発されるシステムであること。
- 利用者（専門家／非専門家）にとって、問題解決の支援または代替を行う能力を持つこと。

本報告書では、実運用型エキスパートシステムを評価の対象とする。実運用型エキスパートシステムは、ユーザの機能要件、性能要件等のニーズを充分満足させることを目標とする。従って、例えば性能要件がユーザに満足されないとすると作り直しを行う必要も生ずる。このようなシステムに対しては、客観的な評価の必要性が高く、本調査研究の対象としても適切である。

一方、エキスパートシステムの有効性が叫ばれ始めた頃には、エキスパートンステムに人工知能技術をどのように適用すへきかの観点でシステムの開発を行っているケースが多かった。つまりシーズ中心の開発である。また、現在においても高度な推論機構を試行し、併せて知識構造を深く究明するンース試行のケースがある。我々はこれを研究開発型エキスパートンステム（ＥＥＳ：Exploring ES）と呼ぶことにする。

実運用型エキスパートンステムと研究開発型エキスパートンステムのめざす目的はたとえ同じ機能を実現していても、それに至るライフサイクルや、要求仕様に対する考え方のアプローチには大きく異なるものがある。特に、評価するときは随分異なる。本報告では研究開発型エキスパートンステムは評価という概念にそぐわないので、その対象としない。

現状では、実運用型エキスパートシステムの多くは、知識ベースと推論機構と

を備えた、いわゆる知識ベースシステムである。しかし、以下の議論で明らかになるように、ここで提案する評価ガイドラインは、知識ベースシステムに限らず、ファジィやニューラルネットを利用するシステムについても一部適用可能なものである。

また、一般に、エキスパートシステムは実用システムに近づけば近づくほど、従来型システムとの統合化が重要となる。それに対して、本報告書で述べる評価ガイドラインは、議論の発散を避けるために、とりあえず、エキスパートシステムとして作られている部分のみを評価対象とする。従来型システムとして最初から作られている部分については、（確立した手法はないものの）従来型の評価手法が適用可能と考えられる。ただし、実運用型システムでは、開発の最終段階において人工知能手法をすてて従来手法で再開発されるケースもある。このガイドラインでは、このような立場から最終的に作りなおされたシステムも実運用型エキスパートシステムと考えて、同じ評価手法を適用するものとする。

2.2 評価に対する基本的な考え方

2.2 1 評価とは

一般に、システムを評価するときの視点としては、社会的な視点、経済的な視点、技術的な視点の3つが存在する。例えば、社会的な視点としては、利用者が知識ベースの知識を基準として納得が出来るのか、信用がおけるのかなどの問題がある。経済的な視点としてはエキスパートシステムがどれたけの利益を生み出すのか、開発の費用効果をどう判断するのかなどの問題が生ずる。これら2つの視点は非常に重要であるが、あらゆるソフトウェアシステムに共通の課題と考えられる。そこで我々は、人工知能技術から直接派生する技術的な諸問題を主な検討の対象とした。これは、技術的な視点からの評価手法が定まっていない現状においては、まず、これを評価ガイドラインとしてまとめることが重要と考えたからである。

評価に関しては、一般的には、表2.2-1のような構造がある。

表2.2-1 評価のレベル

総合的な視点 Evaluation	社会的な視点	Assesment
	経済的な視点	Cost Justification
	技術的な視点	Validation
		Verification

従って、本報告書で述べるエキスパートシステムの評価ガイトラインは、特にシステムの検証（Verification）と妥当性検査（Validation）（これらを総称してV＆Vと呼ぶ）を中心にする。またV＆Vを行う手段としてテスト(Testing)が中心となる［Deutsch 1982］。これらを総称して、本報告書では「評価」という用語を用いることにする。

評価(VV&T) =	検証 Verification	+	妥当性検査 Validation	+	テスト Testing

図2 2-1 評価とは

2.2.2 ＶＶ＆Ｔとは

検証(Verification)とは、システムの内的な正しさを保証する作業、すなわち、システム開発時に生ずるエラーをなくす作業を指し、開発工程において正しくシステムを構築するための確認作業である。このため、ライフサイクルの設定が重要で、それぞれの開発工程で何を行い、何をアウトプットするのかを決めねばならない。開発工程において、上位レベルの設計が、要求仕様の全ての項目を取り込んでいるか、詳細設計の中で上位レベルで記述されているように動くのかを調べることであるとも言える。

妥当性検査(Validation)とは、完成したシステムが（陽には表現しにくい）ユーザの真の要求の目的を満たしているかを確認する作業である。すなわち、妥当性検査では、システムは正確に要求が反映しているのか（正確性）、ユーザにとり有用なものであるといえるのか（有用性）が重要となる。そして、正確性、有用性を確認するためには、通常要求仕様の達成状況を評価することとなる。

テスト(Testing)とは、プログラムの能力を計算機を用いて評価すること、問題領域を設定した範囲内で専門家の知識が知識ベースとして確立されているか否かを評価することである。なおレビューは、一種の静的なテストである。

Ｖ＆Ｖについて、表2.2-2のように様々な定義がある。

表2.2-2　Ｖ＆Ｖの定義

	検証(Verification)	妥当性検査(Validation)	参考文献
1	ソフトウェア開発サイクルの所与の過程での成果が、その前段の過程で確立された要求事項を満たしているかどうかを決定する過程。	ソフトウェア開発過程の最後に、ソフトウェアの要求事項に従っているかどうかを確認するためにソフトウェアを評価する過程。	[ANSI/IEEE 1988]
2	エキスパートシステムの開発工程において、正しい製品を構築する検査である。	エキスパートシステム製品そのものに対して、正しい製品であるのかを検査する。	[Wright 1990]
3	システムを正しく作ること(Building the system "right")	正しいシステムを作ること。(Building the "right" system)	[O'Keefe 1987]
4	定式化された問題が現実の問題をことごとく含み、充分に信頼できる解決策を導出可能なほど充分によく構造化されているかの確証。	エキスパートシステムが特有の問題領域で専門家が保持している知識を正確に表現しているかを決定する過程。	[O'Leary, T.J. 1990]

表2.2-3　V＆Vの定義（つづき）

	検証(Verification)	妥当性検査(Validation)	参考文献
5	ソフトウェアの一貫性、完全性、正確性を決定するため、システムを正しく構築すること。システムの中のエラーを取り除き、それがどんな仕様にも一致することを保証することが目的である。	ユーザニーズや要求に関して、開発プロジェクトから生成された最終プログラムやソフトウェアが正確であることを決定すること。正しいシステムを作ること。システムは何をしているのか、何を知らないのか、何を間違っているのかに関わる。	[O'Leary, D E 1987] [O'Leary, D E 1989]
6	システムの内部の正確さについて言及するもので、構築過程における各々の段階がそれより前の段階で意図されたことを正確に反映していて、内部的な誤りが一つもないということを証明すること。	要求がユーザニーズやシステムの意図するものと一致し、出来上がったシステムが要求を完全に満たしていることを証明すること。検証と妥当性検査との差は、妥当性検査には外部に基準があることである。	[EPRI 1988]
7	エキスパートシステムが正しく開発され、技術的なエラーがないか調べることである。	エキスパートシステムがユーザのニーズを満足することを保証することである。	[Geissman 1988]

2 2.3 V&Vと信頼性、品質との関係

品質は、開発工程の設計の際に作り込むことが重要である。作り込みの品質を大きく分けると、①基本となる当り前の品質（信頼性）と②魅力的な品質（付加価値）になる。前者は、機能の実現の度合を示し、後者はヒューマンインタフェース、保守性の良さの度合などを示す。これらの品質の作り込みを開発工程の中で確かめるのが、V&Vであるといえる。

信頼性は、品質の観点からすると基本となる品質のことである。知識の信頼性、結果の信頼性、時間的に収束していくなど信頼性のレベルは随所にあると考えており、本報告書では、信頼性は一般的な用語として使用する以外は、極力使わないように努力した。

評価の目的は、エキスパートシステムの品質を高めるために行う、その手段としてV&Vを行う。V&Vを行う手段として、実際には、テストが中心になるという関係がある。

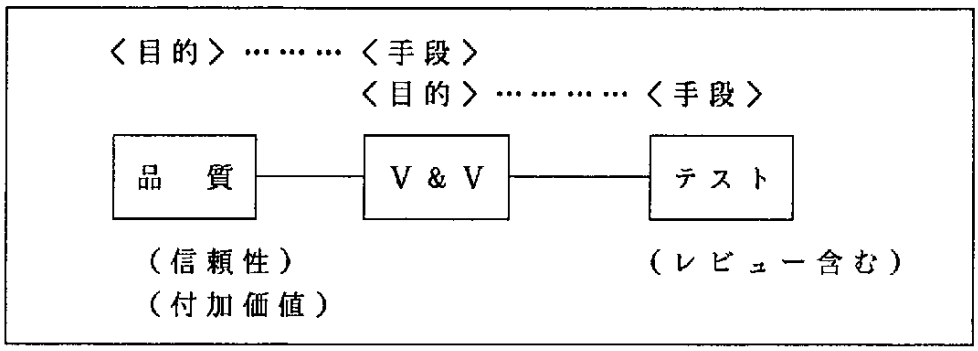


図2 2-2 V&Vと信頼性、品質との関係

2.2.4 評価の難しさ

エキスパートシステムの評価が難しいのは、下記に述べる原因・理由があるからである。

第１の理由は、現段階ではエキスパートシステムのＶ＆Ｖをどのように実施するのかの理解が不足しているため、悪循環に陥っていることによる（図2.2-3 を参照）。すなわち、誰もエキスパートシステムのＶ＆Ｖを要求しないため、Ｖ＆Ｖは行わない、このためＶ＆Ｖのやり方を知らない。Ｖ＆Ｖのやり方を知らないため、誰もＶ＆Ｖを要求しない。現在この悪循環を立つための、Ｖ＆Ｖ方法論が確立していない。

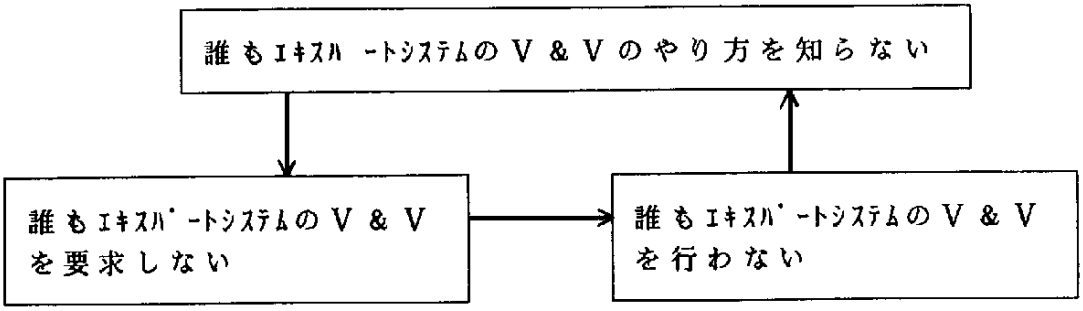


図2.2-3　エキスパートシステムにおけるＶ＆Ｖの悪循環

第２の理由は、ＡＩの分野は特に技術の進歩が激しいため、現状では評価方法の確立までなかなかいきつかない状況にあるためである。評価用ノールや評価方法は、現在は充実していないが、これらは知識獲得をする際、併せて評価も行うという意味で知識獲得のテーマでもある。知識獲得の研究が更に進めば、エキスパートンステムの評価方法もかなり進展する。

第３の理由は、人間のエキスパート（専門家）は、めったに客観的に評価されないことが上げられる。現在、多くのエキスパートシステムは、専門家の持つ知識をもとに構築されており、エキスパートシステムの評価はその専門家を評価することになりかねない。また、自分の知識に基づいて作成されたンステムの評価は、その専門家以外には困難であり、客観的評価とはなり得ない。さらにまた、対象とする問題に対し専門家によって評価の仕方、結果の見方が異なることがある。

第４の理由は、要求仕様が作成されないままに開発が進められていることが多いことである。１つには要求仕様を作成する習慣のない人達がエキスパートンステムを開発していることもある。また最初に作成した要求仕様とはほど遠い内容にシステムが変化してしまうので要求仕様を作成する意味がないと考えている人達もいる。ドキュメントの種類、内容、形式、及びトキュメント技法が不明確なのでレビューが困難である。このように要求仕様や他のトキュメントが曖昧、またはないため妥当性検査(Validation)をする際、形式的に評価することが困難な状況にある。

第５の理由は、要求仕様や設計仕様の変更が頻繁に起こるため、テスト計画を立てにくいことである。再現性のあるテストデータや要求仕様からテスト可能なテストデータを見つけにくい。またテスト結果を評価するとき、広く受け入れられ、信頼されている方法がないなど、テストするための評価方法や環境が確立し

ていないことがある。
そのほか、評価が難しい原因は、エキスパートシステム自身に内在している、①普遍的でない経験的知識を扱う、②推論により何処まで解法できるのか予測がつかない、③曖昧な情報も取り扱う、④学習機能によりシステムが成長する、⑤問題解決の対象とする領域が明確でない、⑥網羅的なテストはしにくい、⑦すへて正解でなくとも使用できる等が上げられる。

以上、評価をする上でこれらの困難はあるが、評価方法を確立する要求は強い。

その第1の理由は、ほとんどがプロトタイプ版とはいえ、多くのエキスパートシステムが世の中に出現している事実である。そして、どのようなエキスパートシステムが実際に役に立っているのかを判断する基準の指標作りの必要性である。
第2の理由は、今後原子力や航空管制等いわゆる超高性能な信頼システムに対しても、エキスパートシステムの適用が考えられ始めており、その前に評価に対する材料をできるだけ世の中に提供することが必要と考えたからである。
第3の理由は、エキスパートシステムのモジュラリティの高さを生かす評価手法が考えられるからである。エキスパートシステムは「モジュラリテイが高い」のでドキュメントなしでソースコードをそのまま読むことが出来るなど、ある一面では評価が容易であることも指摘されているからである。

2 2.5 知識獲得活動とV&V

エキスパートシステム開発には、知識、すなわち、問題解決に必要な情報を、専門家などの知識源から収集・定式化し、それを人工知能技術が提供する手法で取扱える形に変換し、さらにコンピュータで扱えるように構造化する作業が必要である。また、エキスパートシステムは一度動きだしてからも、知識の誤りや矛盾、あるいは、知識の増大に対応して、常に、変更され続けるのが普通である。知識獲得活動は、知識の収集・知識の変換・知識の構造化・知識の洗練化の4段階に分けることができる[Kobayashi 1990]。

知識獲得活動とエキスパートシステム開発のライフサイクル、ならびに、V&V作業との関連は図2.2-4のように表現できる。
すなわち、エキスパートシステム開発作業が上流から下流に進むにつれて,知識の収集・知識の変換・知識の構造化・知識の洗練化の順に中心となる作業が変化していく。従来型システムの開発では、通常、ライフサイクルの各フェーズが終了した時点でV&Vの作業が行われるが、エキスパートシステムの場合は、各フェーズで、獲得した知識をプロトタイプによって実行することが比較的容易である。この意味で、知識獲得活動に関する議論はすへて、V&Vと深く関連する[Gains 1990],[Dhaliwal 1990],[Hart 1989]。

知識がない初期状態からの知識の収集・獲得作業は、通常、開発担当者と専門家とのインタビューによって行われ、発想と体系化の繰り返しによって問題の性質が明確化される。この際のV&Vは、問題定式化の方法が概ね妥当てあるかど

うか、既存技術と知識処理技術との役割分担、統合化が明確になっているなどの観点から行われる。開発担当者と専門家との意思疎通を充分に行うために、対象分野に対する共通のドメインモデルと知識処理に対する共通のタスクモデルをもつことが望ましい。また、この段階を支援するツールは、原理的に、知識の整理・発想の支援といった面で利用できる。従って、システム要求を開発初期段階において明確にすることが出来る点において、妥当性検査に有効である。

知識の変換・構造化の段階では、専門家が明文化した知識、並びに、対象分野の対象モデル・教科書的知識を現在の知識処理技術を用いてコンピュータ上で操作できる形式に変換する作業が中心となる。コンピュータ上で操作可能になる前の知識は、通常、散逸しており、その整合性は保証されないと考えられる。従って、この段階のＶ＆Ｖ作業では、十分な領域モデルの蓄積と整合性のチェックが重要な問題となる。単純な分類問題については、この段階で専門家を支援する知識獲得支援ツールもいくつか発表されている。このようなツールでは、各分野の専門家が知識技術者の援助なしに、ツールと対話処理を行っていく過程で、短期間に質の高い知識ベースを構築することに重点がおかれている。これは、獲得した知識の検証に役立つ。

知識の洗練化の段階では、知識ベースの一貫性を保持すると同時に、知識ベースに誤りや矛盾が生じた場合にそれを修正する作業が重要である。この段階では、コンピュータのサポートなしにＶ＆Ｖの作業を行うことは難しい。この範疇にはいる代表的な知識獲得支援ツールとしては、いわゆるハイブリットツールの知識エディタやデバッガが含まれる。これらは、知識の妥当性検査・検証に利用できる。

現在のところ広い分野・範囲で利用できる知識獲得支援ツールは非常に少ない。そのために、以下で述べる評価ガイドラインでは、Ｖ＆Ｖと知識獲得支援ツールとの関連性は重視せず、かわりに、様々なテスト手法を分類して、評価項目と関連づけている。しかし、今後はＶ＆Ｖの側面を重視した知識獲得手法・知識獲得支援ツールが次第に増えて行くものと考えられる。

《参考文献》

[Dhaliwal 1990] Dhaliwal, J.S., Benbasat, I A Framework for the Comparative Evaluation of Knowledge Acquisition Tools and Techniques Knowledge Acquisition, Vol 2, No 2, pp 145-166,1990

[Gains 1990] Gains, B.R., Show, M L. Foundations of Knowledge Acquisition. Proc. JKAW'90, pp.3-24, 1990

[Hart 1989] Hart,A. Knowledge Acquisition for Expert Systems Kogan-Page, 1989

[Kobayashi 1990] 小林重信,寺野隆雄（編）：知識システムハントブック オーム社, 1990.

[Prerau 1990] Prerau,D.S Developing and Managing Expert Systems Addison-Wesley, 1990.

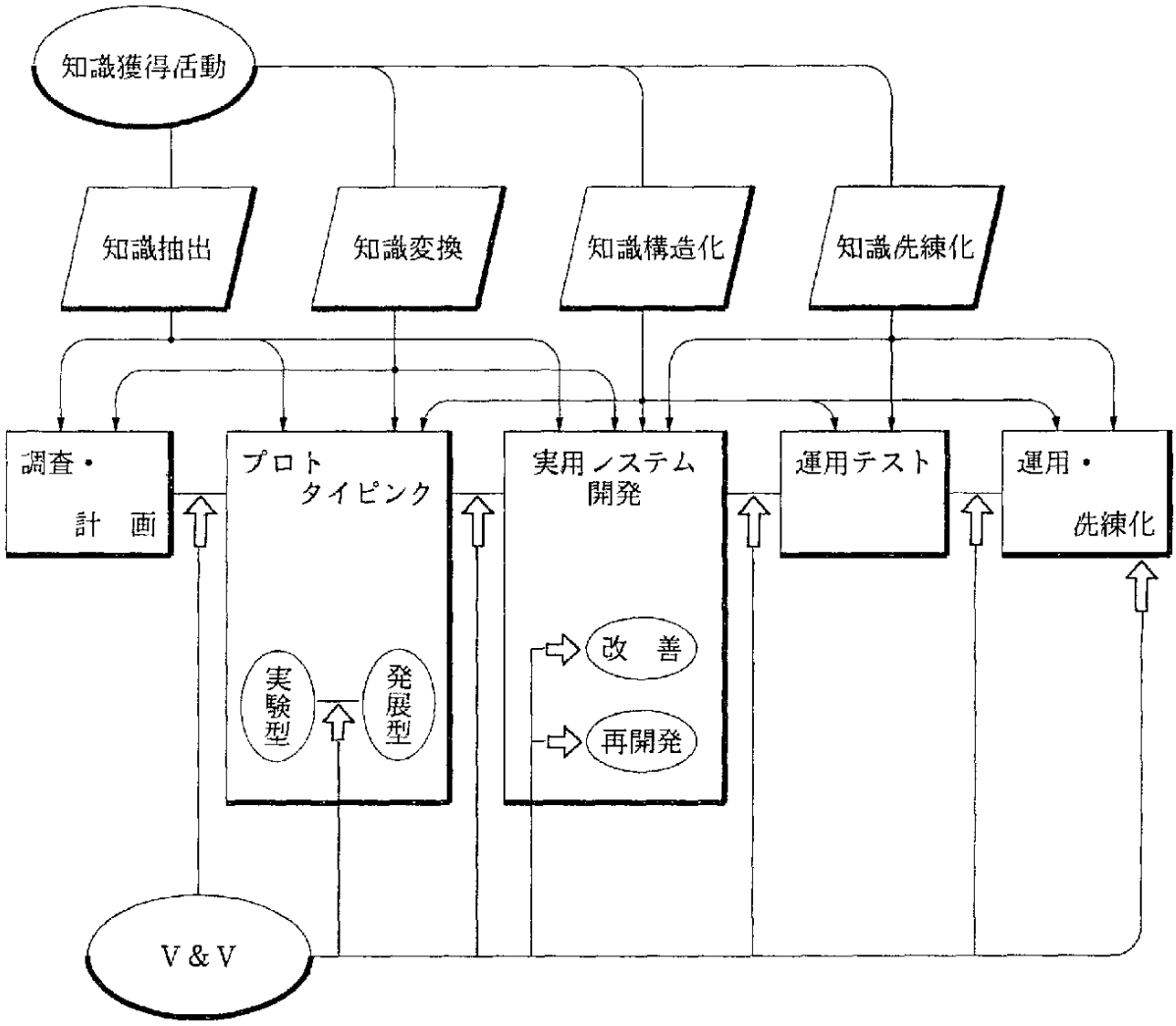


図2 2－4　知識獲得活動と開発フェーズ・V&Vの関係

2.3　評価体系の基本的な考え方

我々の設定した評価の体系は図2.3-1の通りである。

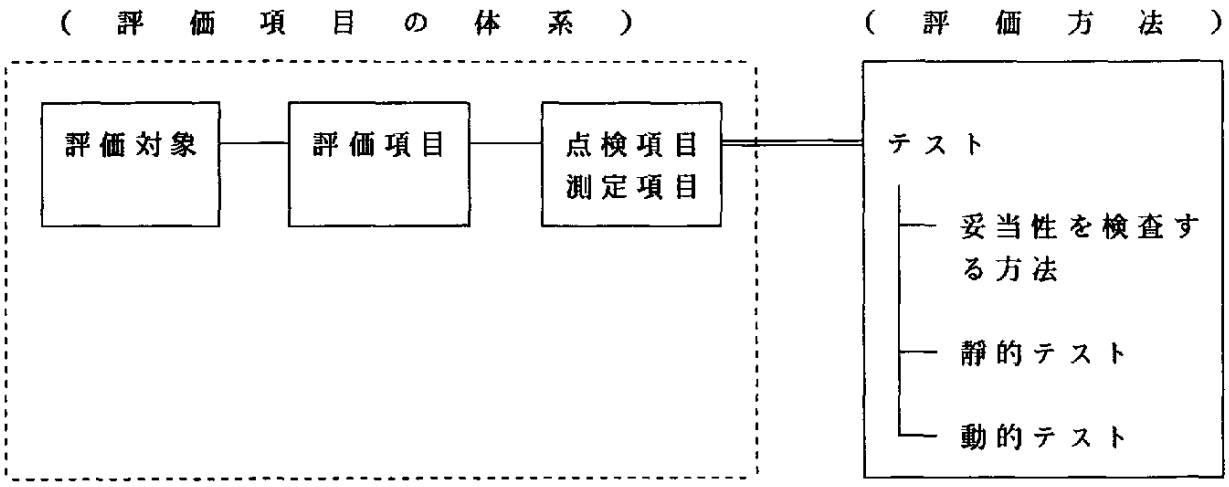


図2.3-1　評価の体系

品質管理の分野で品質の保証をする際、「品質機能展開」のアプローチが用いられ、要求された品質の満足や開発の効率化に役立てられている。我々は、この「品質機能展開」の手法を用いて、品質を保証する対象（評価対象）を定め、評価対象をブレークダウンして、評価項目を設定し、更に点検項目、測定項目に展開する。この様に階層構造化され、細分化された項目を「評価項目の体系」と呼ぶことにする。この項目を実施する手段が「評価方法」である。評価方法とは、いわゆるテストをすることであり、マシンを用いてのテストばかりでなく、人手によるレビューも含めるとする。

なお、V＆Vとの関係を述べると、評価項目の体系とはV＆Vを実施する目標であり、V＆Vを実施する手段としてマシンによるテストやレビューをするという位置付けである。

2 3.1　評価対象

今回、主に技術的な観点からエキスパートシステム自体を評価することとしているが、その場合の評価の対象は、基本的には計算機システムの考え方により、①システムへの要求、②その結果、及び③その処理の仕方、④システム全体に係わること、及び⑤保守の問題に大きく５つに分けて検討することとする。
これを図に示すと下記のようになる。

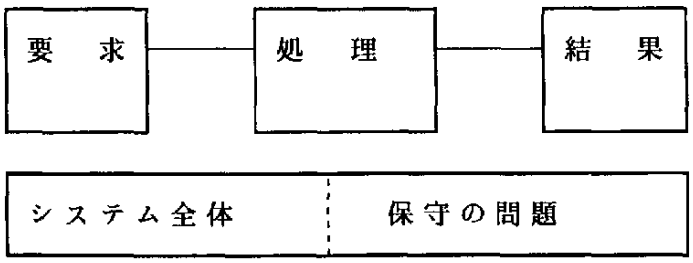


図2 3-2　評価の対象の枠組み

処理の仕方は、さらにエキスパートシステムの枠組みから①知識ベース、②問題解決・推論機構、③ヒューマンインタフェーズの3つに分ける。以上を整理して、我々は7つの評価の視点（評価対象）を設定した。これを図2 3-3に示す。

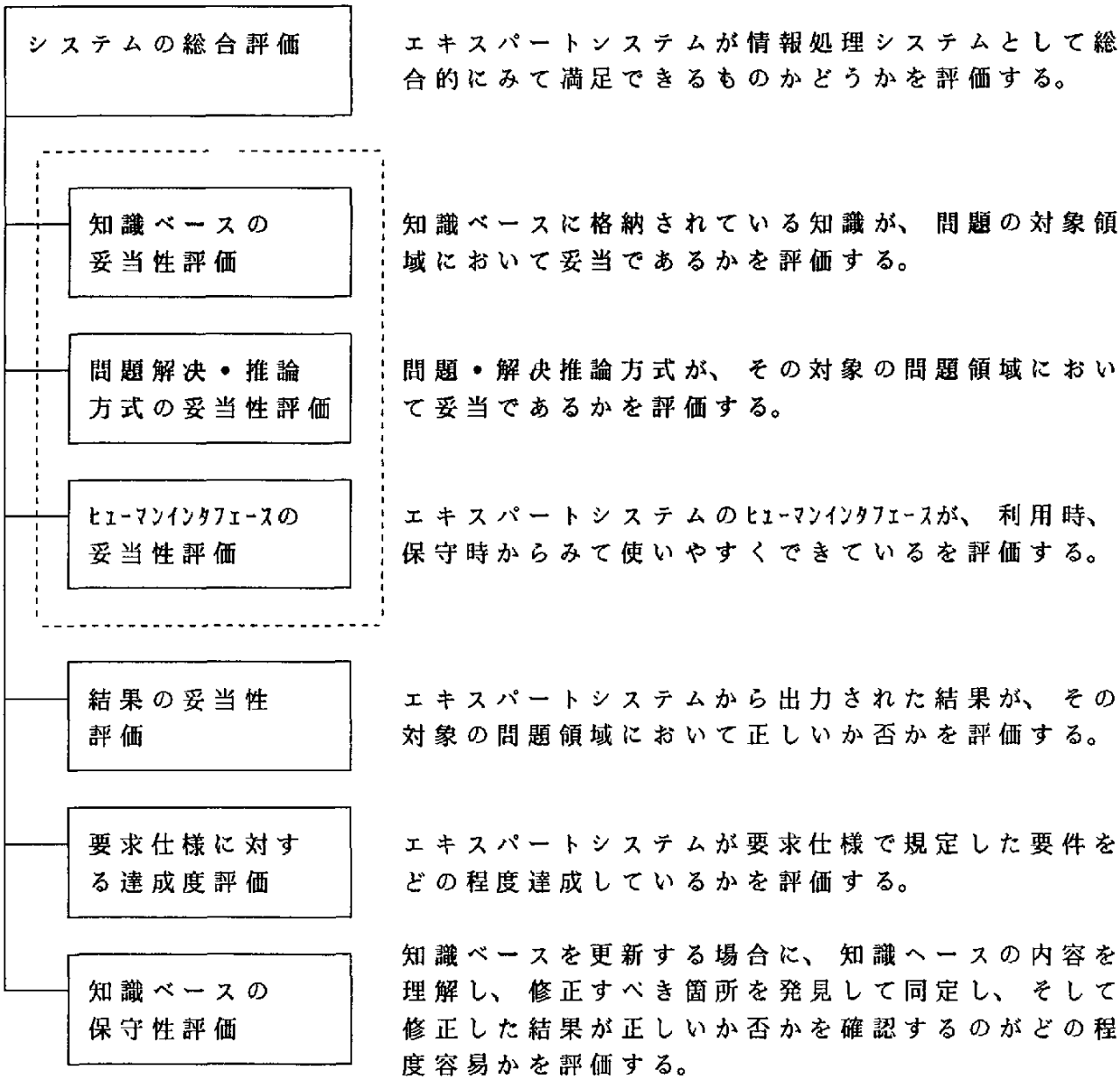


図2 3-3　評価対象（7つの評価の視点）

2 3.2 評価項目

７つの評価対象をさらにブレークダウンした項目を「評価項目」と呼ふ。評価項目を吟味すると結果に関する評価（結果系）と要因に関する評価（要因系）とその他経済性・社会性に関する評価の３つに大きく分類される。要因系は更に、評価方法そのものを評価する場合と方法論・技法・ノールの評価の場合に分けられる。以上の考えを整理すると次のようになる。

表2.3-1　評価項目の共通的な分類

結果系	P	プロダクトに関する評価
要因系	E	評価方法に関する評価
	M	開発／保守方法・技法・ノールに関する評価
その他	O	経済性、社会性に関する評価

この共通な考え方に基づき、評価項目を分類・整理すると表2 3-2のようになる。

表2 3-2　評価項目

評価対象	分類	評価項目
1　知識ベースの妥当性評価	P	1-1　知識ベースの正当性評価 1-2　知識ベースの完備性評価 1-3　知識ベースの一貫性評価
	E	1-4　経験的知識を正当化する方法とその確認
	M	1-5　知識表現の選択と適切さの確認 1-6　知識獲得方法の選択と適切さの確認
2. 問題解決・推論方式の妥当性評価	P	2-1　推論処理の妥当性評価 2-2　推論時間の妥当性評価
	M	2-3　推論手法の選択と適切さの確認 2-4　不確実情報に対する処理技法の適切さの確認 2-5　学習機能の適切さの確認
3　結果の妥当性評価	P	3-1　正解の評価 3-2　解の網羅性評価 3-3　解の最適性評価
	E	3-4　解の評価方法の適切さの確認

表2 3-3 評価項目 （続き）

評価対象	分類	評価項目
4. 要求仕様に対する達成度評価	P	4-1 要求仕様内容に対する評価 4-2 要求仕様に対する達成度評価
	M	4-3 要求仕様設定手段の確認
5 知識ベースの保守性評価	P	5-1 変更箇所発見の容易性評価 5-2 知識ベースの理解性評価 5-3 変更箇所の局所性評価 5-4 知識ベースの統一性評価
	M	5-5 品質安定性の確認方法の確認 5-6 保守マニュアルの整備状況の確認 5-7 保守用ノールの整備状況の確認
6 ヒューマンインタフェースの妥当性評価	P	6-1 利用時のインタフェース評価 6-2 保守時のインタフェース評価
7 ノステムの総合評価	P	7-1 テーマ選定の妥当性評価 7-2 利用者の満足度評価 7-3 統合性の評価 7-4 成長可能性の評価 7-5 開発／保守用トキュメントの評価
	M	7-6 開発方法の適切さの確認 7-7 ノール・言語選択の適切さの確認
	O	7-8 ノステム開発の生産性評価 7-9 ノステムのコストパフォーマンス評価 7-10 セキュリティ対策の評価

2.3 3　点検項目／測定項目

評価項目を更に展開したものが点検項目と測定項目である。

点検項目とは、チェックリストのことである。システム構築におけるデザインレビューの際、または構築後の評価・点検の際に用いる。
点検項目を抽出分類する際、特にcriticalな項目となるようにしたこと、他の評価項目に属する点検項目と重複がないようにしたこと、説明調として分かりやすくしたこと、点検項目間でグループ化が出来るものはグループ化したことなどに配慮して作成した。

測定項目とは、実際に定量的に測定が出来る項目のことである。測定項目は、数式を用いて表現したものが多い。

2 3.4 テスト

(1)概　要

ここでは、まず最初に従来のソフトウェアに対するテストを概観した後で、エキスパートシステムの検証および妥当性検査（V&V. Verification and Validation）を行う際に使用可能で有用であると思われるテスト方法（確認手段）を網羅的に分類して紹介する。（詳細説明は3.6「評価に用いられるテスト方法」を参照のこと。）

評価に際しては、評価対象のエキスパートシステムの重大度（リスク、安全性の面）・特質や開発・評価コストなどを勘案して、ここで紹介されているテスト方法を適宜選択し、テストをする必要がある。

以下で紹介するテスト方法の分類体系・次元は半入れ子構造になっているので、明確にするために分類番号を付与してある。

なお、ここに紹介されているすべてのテスト方法を任意に組み合わせて選択して、エキスパートシステムのテストに使用出来るわけではなく、この半入れ子構造を意識して選択する必要がある。

(2)ソフトウェアのテストとエキスパートシステムのテスト

ここでは、まず最初に、エキスパートシステムのテストについて説明するための導入部として、従来のソフトウェアに対してどういうテストが行われ、どのような手順（テストサイクル）でテストが行われているかを簡単に紹介する。

図2 3 4-1に示すように、一般に、ソフトウェアの開発では、ウォータフォールモデルのV字型開発サイクルを採用しており、開発サイクル前半（V字の左半分の部分）の品質の作り込み過程である「要求分析・仕様作成」、「基本設計」、「詳細設計」の各フェーズでの成果物である各種の仕様書・設計書に対して、レビュー、ウォークスルー、インスペクションなどの人手による静的テスト手法を用いて、その正当性や一貫性を確認している。そして「コーディングフェーズ」での成果物であるプログラムモジュールに対しては、静的分析テストツールを使用して、構文的あるいは論理的な誤りを検出したり、人手によってプログラムの1行1行に対して、その正当性や一貫性を精査するコードウォークスルーやコートインスペクションが行われている。

開発サイクルの後半（V字の右半分の部分）では、前半のサイクルの逆順にほぼ対応した、動的テスト手段を使用する「単体テスト」、「統合テスト／インタフェーステスト」、「システムテスト」の各フェースがそれぞれ実施される。

ソフトウェアテストに関する最近の傾向は、動的テストではプログラムに対する入力データのすべてのケースをテストするのは時間的にもコスト的にも不可能であるので、静的テスト、すなわち、人手によるレビュー、ウォークスルー、インスペクションや、静的分析テストツールによるプロクラムの分析をより重視するようになってきている。そして、ソフトウェア信頼度成長モデルを使用して、ソフトウェアの信頼度を予測したり、テストの打ち切り時期を決めている。テストデータに関しても、従来のように仕様書や設計書に基づいて作為的にテストデータを選定するのではなく、運用プロフィールを反映した、あるいは統計的に見て妥当なテストデータを適宜選定・使用してテストをするようになってきている。

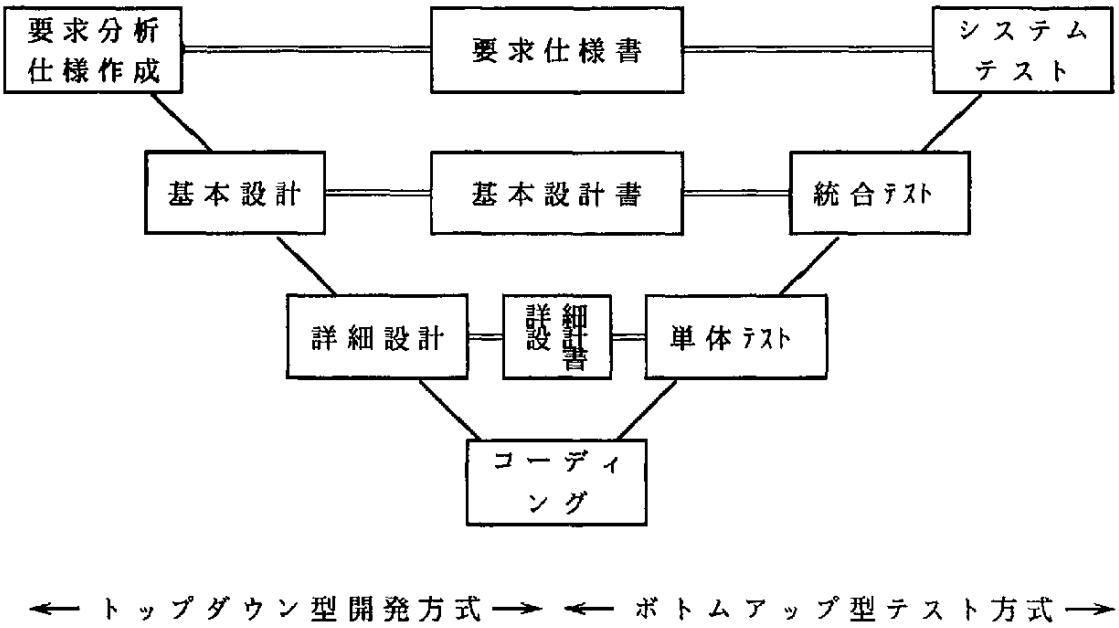


図2.3.4-1 ソフトウェア開発のウォータフォールモデル
（V字型開発サイクル）

ソフトウェアのテストは次の手順で行われている。

①テストの計画
②テスト項目の設計
③テストプログラムの設計
④テストプログラムのコーディング
⑤テストの実施
⑥テストの報告

この一連のテストの手順の考え方は、エキスパートシステムのテストに際してもほぼそのまま適用可能である。

エキスパートシステムでは、その開発にプロトタイピングを採用しており、どのフェーズでも、不完全ではあるがそれ自体（単体）で動くプロトタイプが存在しており、従来ソフトウェアの単体テストで必要とするドライバやスタブは必要がないので、③や④の作業は必要ないか、あるいはかなり軽減される場合が多い。また、エキスパートシステムはインクリメンタルに機能が増加して行くので、単体テストというイメージはない。

(3)エキスパートシステムのテスト方法の分類体系・次元

エキスパートシステムのテスト方法は、表2 3 4-1に示すようにな次元で分類される。

表2 3 4-1 エキスパートシステムのテスト方法の分類体系・次元(その1)

次元X: 妥当性を検査する方法による分類
X-1: 内容的妥当性テスト X-2: 基準に基づく妥当性テスト X-3: 構成概念的妥当性テスト X-4: 表面的妥当性テスト X-5: 未来予測的妥当性テスト X-6: 併存的妥当性テスト X-7: 過去予測的妥当性テスト
次元Y/Z: テスト対象を実行するか否かによるテスト方法の分類
Y : 静的テスト/分析 Z : 動的テスト
次元Y: 静的テスト/分析でのテスト手段による分類
Y-1: 人手による Y-1-1: レビュー Y-1-2: ウォークスルー Y-1-3: インスペクション Y-1-4: 変則評価 Y-1-5: リスク分析 Y-1-6: ハザード解析 Y-1-7: 故障の木解析(FTA) Y-2: 静的分析ツールによる Y-2-1: ルールベースの論理的一貫性分析 Y-2-2: ルールベースの論理的完備性分析 Y-2-3: ルール間の相互参照分析 Y-2-4: 確信値の矛盾検出 Y-2-5: データフロー分析 Y-2-6: 制御フロー分析

表2.3 4-1　エキスパートシステムのテスト方法の分類体系・次元（その２）

次元Z：動的テストでのテストサイクルなどによる分類項目
Z-1：テストデータ（問題）の選定方法による分類 Z-2：テスト対象の内部構造を見て、その結果に基づくか、あるいは外部から見た機能に基づくかでのテスト方法の分類 Z-3：同じテストデータを再度使用してテストをするか否かによるテスト方法の分類 Z-4：テストを行う場所でのテスト方法の分類 Z-5：特定の目的に特化したテストでのテスト方法の分類 Z-6：推論過程の正しさを調べる手段による分類 Z-7：テスト結果の比較対象／手段による分類 Z-8：評価の正しさを評価するための手段による分類 Z-9：評価者の偏向を取り除く手段による分類
次元Z-1：テストデータ（問題）の選定方法による分類
Z-1-1：ランダムデータ法 Z-1-2：同値分割法 Z-1-3：限界値分析法 Z-1-4：原因ー結果グラフ法 Z-1-5：層化抽出法 Z-1-6：誤り推定法 Z-1-7：運用プロフィール法 Z-1-8：実データ法（本番データ法） Z-1-9：標準テスト問題法（能力レベル別） Z-1-a：テスト網羅計測法（テストデータの質の評価法） Z-1-b：変異テスト法（テストデータの質の評価法）
次元Z-2：テスト対象の内部構造を見て、その結果に基づくか、あるいは外部から見た機能に基づくかでのテスト方法の分類
Z-2-1　ホワイトボックステスト／構造テスト Z-2-2：ブラックボックステスト／機能テスト
次元Z-3：同じテストデータを再度使用してテストをするか否かによるテスト方法の分類
Z-3-1：回帰テスト Z-3-2：新規テスト

表２３４-1　エキスパートシステムのテスト方法の分類体系・次元（その３）

次元Z-4：テストを行う場所でのテスト方法の分類
Z-4-1：フィールドテスト Z-4-2：実験室テスト
次元Z-5：特定の目的に特化したテストでのテスト方法の分類
Z-5-1：負荷テスト Z-5-2：感度分析テスト Z-5-3：削除テスト
次元Z-6：推論過程の正しさを調べる手段による分類
Z-6-1：説明機能 Z-6-2：プロトコル分析（内観分析）
次元Z-7：テスト結果の比較対象／手段による分類
Z-7-1：過去／既存のデータ／結果 Z-7-2：１人の専門家 Z-7-3：専門家のパネル（複数の専門家） Z-7-4：対照基準グループ Z-7-5：併行運用 Z-7-6：シミュレーション Z-7-7：他のモデル Z-7-8：実　験 Z-7-9：実地調査
次元Z-8：評価の正しさを評価するための手段による分類
Z-8-1：専門家間の一致係数 Z-8-2：信頼区間 Z-8-3：第Ｉ種の誤り Z-8-4：第Ⅱ種の誤り
次元Z-9：評価者の偏向を取り除く手段による分類
Z-9-1：チューリングテスト／目かくし評価 Z-9-2：クロスチェック Z-9-3．統計的手段

(4)テスト方法の選択例

テストおよび評価を行うに際しては、その目的・目標に応して、適宜テスト方法を選択して使用する必要がある。

以下に、適用領域や適用業務の致命度あるいは評価の必要度合の順に、妥当な組合せの選択の例を示す(軽重順)。

[ケース1]

X-1: 内容的妥当性テスト

Y : 静的テスト/分析

Y-1: 人手による

Y-1-2: ウォークスルー

このケースでは、例えば、知識ベースに格納されている知識そのものが正しいかを、ウォークスルーで確かめる。

[ケース2]

X-1: 内容的妥当性テスト

Y : 静的テスト/分析

Y-2: 静的分析ツールによる

Y-2-1: ルールベースの論理的一貫性分析

Y-2-2: ルールベースの論理的完備性分析

このケースでは、知識ベースに格納されている知識の一貫性と完備性に問題ないかを検査するために、これらのチェック機能を保持している静的分析ツールを使用し、その一貫性と完備性を確かめる。

[ケース3]

X-4: 表面的妥当性テスト

Z : 動的テスト

Z-1-8: 実データ法(本番データ法)

Z-2-2: ブラックボックステスト/機能テスト

Z-4-1: フィールドテスト

Z-7-5: 併行運用

このケースでは、エキスパートシステムの能力を調べるため、あるいは、実際に実務に使用できるかを調べるため、フィールドで、実際に実務で使用している実データ(本番データ)をエキスパートシステムに入力して、ブラックボックステスト/機能テストを行う。出力された結果と、同じ実データ(本番データ)を既存の併行運用されているシステムに入力し、処理させた出力結果とを比較評価する。

[ケース4]

X-6: 併存的妥当性テスト

Z : 動的テスト

Z-1-2: 同値分割法

Z-2-2: ブラックボックステスト/機能テスト

Z-7-2: 1人の専門家

Z-8-3: 第I種の誤り

Z-8-4: 第II種の誤り

このケースでは、同値分割法を使用してテストデータ(問題)を選定し、そのテストデータ(問題)をエキスパートシステムに入力して、ブラックボックステスト/機能テストを行う。出力された結果と、同じテストデータ(問題)に対し

て専門家が解いた答とを比較評価する。比較評価の際に、エキスパートシステムの結果が正しいのに誤っている、あるいは、逆に誤っているのに正しいという誤判定がないかを綿密に確認する。

［ケース５］

X-2：基準に基づく妥当性テスト

Z　：動的テスト

Z-1-1：ランダムデータ法

Z-2-2：ブラックボックステスト／機能テスト

Z-7-7：他のモデル（ＯＲモデル）

このケースでは、計画問題用のエキスパートシステムの能力を調べるため、新たにＯＲモデルに基づいたシステムを構築する。この場合の基準はＯＲモデルシステムの答である。ランダムデータ法を使用してテストデータ（問題）を選定し、そのテストデータ（問題）をエキスパートシステムに入力して、ブラックボックステスト／機能テストを行う。出力された結果と、同じテストデータ（問題）をＯＲモデルシステムに入力し、処理させた出力結果とを比較評価する。

［ケース６］

X-6：併存的妥当性テスト

Z　：動的テスト

Z-1-3：限界値分析法

Z-2-1：ホワイトボックステスト／構造テスト

Z-6-1：説明機能

Z-6-2：プロトコル分析（内観分析）

Z-7-3：専門家のパネル（複数の専門家）

Z-8-1：専門家間の一致係数

Z-9-2：クロスチェック

このケースでは、限界値分析法を使用してテストデータ（問題）を選定し、そのテストデータ（問題）をエキスパートシステムに入力して、ホワイトボックステスト／構造テストを行う。その際に、エキスパートシステムの推論過程を説明機能を用いて調べる。一方、同じテストデータ（問題）に対する専門家の推論過程もプロトコル分析（内観分析）を用いて調べる。エキスパートシステムの推論過程と専門家の推論過程とを、複数の専門家によって比較評価する。各専門家の評価結果がどの程度一致しているかを一致係数を用いて検定したり、あるいは、評価結果を相互に交換し、各専門家は再度、各評価結果が妥当かを吟味する。

２．４　要求仕様に対する基本的な考え方

２４.１　要求仕様は作成できるか

現在、エキスパートシステムの開発に当たり、要求仕様を予め明確化し、作成して開発を進めていることは多くはない。この理由は以下のように考えられる。

第１に、エキスパートシステムの対象分野は問題の構造が悪構造であると同時に問題自体を明確に規定することが難しい悪定義問題であると言われている。悪定義な問題とは、対象とする範囲や問題解決のためにどのような回答を出せば良いのか等が開発当初においては明確に定義付けることが困難な問題のことである。また例え要求仕様を作成したとしても、すぐ要求が変更されてしまい作成する意味がないと言うこともある。従って、開発当初においては要求仕様を作成することは出来ないとする考え方がある。

第２に、従来のソフトウェア開発で要求仕様が作成できるのは、ユーザの要求を一旦凍結して開発の作業へ移行するからである。しかしエキスパートシステムの開発は逐次的な実験指向の傾向が強く、ユーザの要求を一旦凍結することが出来にくい。

第３に、エキスパートシステムの開発工程は、大筋では計画、分析、設計、開発、保守の工程は変わらない、ということに対する認識が薄いことにある。このためレビューや開発管理も徹底していない。現状ではエキスパートシステム開発における要求仕様化技術が未発達である。

これに対し、一方、エキスパートシステムといえども一つのシステムであるため、どのような機能を果たし、どの程度の性能のシステムが欲しいかが明確でないままにシステム開発を開始すること等はありえないとする立場である。近年、実用に耐え得るエキスパートシステムが出現しているが、これらの多くは、この立場を我々は、実運用型エキスパートシステム（ＯＥＳ）と呼んでいる。

ＯＥＳにとっては、品質要件をシステムに作り込むことが重要である。このためには要求仕様書を作成する。すなわち、Ｖ＆Ｖの鍵は要求仕様を作り込むことにあると言える。要求仕様書の重要な理由は、以下の通りである。

①要求仕様はそのシステムを評価する基準のベースになる。

②2.5節で述べるライフサイクルのモデルで要求仕様を明確化できる。

③客観的な第３者による評価を行うことが出来る。

24.2 エキスパートシステムの要求仕様の特徴

我々のプロトタイピングアプローチでは、開発の各フェーズごとに最初の作業として要求仕様を定義する。このもとでシステムを開発し、要求仕様に記述された要件を満足しているのかを検討し、次の開発フェーズへ進む。各段階で作成される要求仕様は、評価と共に改訂され、また次第に詳細化される。そして、各段階における要求仕様は、その段階で作られるシステムに対するV&Vの評価の基盤として利用される。

要求仕様を作成する作業においては、以下の点に留意して行う。

- エキスパートシステム開発では、適切な対象業務範囲の決定が重要なポイントであるため、実験型プロトタイピング開発にはいる直前の企画提案書が最初の要求仕様書に担当するものとして重要である。
- システム化に先立ち、開発者とユーザはシステムに対する要求を整理し、要求仕様作成のために出来る限りの努力をすること。
- 要求仕様を記述するために、業務の内容・性質、システムに望む性能・機能、最終ユーザのレベルなどを元に検討されたシステムに対する妥当性の基準やシステムの出力する解の満足度などに関して事前に整理を行う。
- 業務には行うべきことや行ってはいけないことがあるので、これをチェック項目や業務における制約条件として要求仕様を作成する際に利用する。
- 要求定義方法は既存のものを用いることが可能であるが、開発者とユーザによる積極的なレビューを繰り返すことが必要である。

2.5 ライフサイクルの基本的な考え方

2.5.1 なぜプロトタイピングアプローチか

プロトタイピングアプローチをとる第1の理由は、エキスパートンステムの対象とする分野の問題解決方法は、従来の定型業務のように決定的な処理手順て表現することが難しいケースが多く、非決定的な処理にならざるを得ないことにある。このような問題構造を一般に悪構造（Ill Structured）とよんでいる。悪構造問題をウォターフォールアプローチでトップダウンに開発することは難しい。このため、部分的に少しづつ作成し問題構造を明確にしていきながら開発を進めるプロトタイピングアプローチが適している。

第2の理由は、エキスパートシステムが扱う問題は、悪定義（Ill Defined）であるからである。このため、開発の最初は利用者の要求仕様が明らかではないが、プロトタイピングを繰り返すことにより、利用者の要求仕様が次第に明確化・洗練化されるという立場をとるからである。

我々の提案するライフサイクルモデルは、プロトタイピングアプローチに基づくウォーターフォールアプローチである。各フェーズが終了することに、直前の段階で定義された要求仕様に対するV＆Vが実施され、次の開発フェーズへむけての仕様が決定する。

もちろんプロトタイピングの考え方は、従来型ソフトウェアの開発においても成立するものである。エキスパートシステムの場合には、特に、

① 問題解決に必要な情報（知識）とそれに必要なアルゴリズムの性質が開発初期には明確になっていない点、

② システムが成功をおさめるには、継続的な保守・拡張の作業が不可欠となる可能性が強い点

の2つが特徴的である。

しかしプロトタイプによるアプローチは、ユーザに過大な期待をいたかせる危険性があるので運用上充分注意する必要がある。

一般的にプロトタイプを作成する目的として三つの立場が考えられる。

① 技術的に未経験な部分、不安な部分を切り出して、その部分の簡単なモデルを作成して技術的可能性を検証する。

② システムの技術ポイントを抽出するために、既知の部分から作成し、不明な部分はブラックボックス化しンステムイメージを構築する。

③ ンステムのイメージを共有化しンステム運用時に仕様上のトラブルを防止するために、システムの動作イメージが見えるようなプロトタイプを作成する。

悪構造の問題に対応するという観点でプロトタイプアプローチを見た場合、①の立場で開発する場合が主流となる。

また、研究的要素が強いエキスパートンステムについては、ライフサイクルという概念をあてはめにくいという意見も存在する。これについては、初期的な要求仕様をまとめるまでの試行的な研究開発を、特に、探査型プロトタイプと呼び、探査型プロトタイプについては、要求仕様が与えられないために、V＆Vの対象とはしないこととした。

2 5 2　エキスパートンステム開発のライフサイクル

エキスパートンステムの対象分野は問題の構造が不明確な、悪構造問題であり、同時に問題自体を明確に規定することが難しい悪定義問題である。従ってンステムの開発当初においては要求を明確に定義付けることが困難で、変更が頻繁に行われることとなる。この様なことを考慮したエキスパートンステム開発のライフサイクルが、表2.5-1のように各種提案されている。

表2 5-1　エキスパートンステム開発のライフサイクル

[Miller 1989]	[Kobayashi 1989]	[Buchanan 1983]	[Bobrow 1986]
ﾌﾟﾛﾄﾀｲﾌﾟ ﾝｽﾃﾑ 要求の獲得 ﾄｯﾌﾟ ﾚﾍﾞﾙ設計 初期ﾌﾟﾛﾄﾀｲﾌﾟ 初期ﾌﾟﾛﾄ評価 ﾍﾞｰｽﾗｲﾝｼｽﾃﾑ 要求仕様 詳細な知識の獲得 主要な設計変更 ﾍﾞｰｽﾗｲﾝﾉｽﾃﾑの開発 ﾍﾞｰｽﾗｲﾝﾝｽﾃﾑの評価 ﾉｽﾃﾑの統合 ﾃｽﾄと統合化 最新ﾉｽﾃﾑ	ｼｽﾃﾑｱﾅﾘｼｽ 問題の設定 既存技術の評価 知識源の同定 ﾌﾟﾛﾄﾀｲﾋﾟﾝｸﾞｱﾅﾘｼｽ 専門家ﾓﾃﾞﾙ同定 ﾕｰｻﾞﾓﾃﾞﾙの同定 知識ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ 知識表現の選択 知識の移植 保守・運用 知識ベース管理 性能の評価	定義 概念化 定式化 ｲﾝﾌﾟﾘﾒﾝﾄ テスト	定義 概念化 プロトタイピング ﾕｰｻﾞｲﾝﾀﾌｪｰｽ テストと再定義 知識ベースの更新
[Keller 1987]	[Martin 1987]	[富士通 1988]	[日立 1987]
調査 構造化分析 知識獲得 ﾊｰﾄﾞｳｪｱ分析 設計 知識獲得 知識ベース設計 ｲﾝﾌﾟﾘﾒﾝﾄ 許容テスト ﾉｽﾃﾑ統合化	要求分析 モデルの定義 ｱｰｷﾃｸﾁｬの設計 ｲﾝﾌﾟﾘﾒﾝﾄとテスト 評価	調査 計画 プロトタイピング ステージ0 ステージ1 ステージ2 公開準備 運用テスト 運用	ンステム分析 知識予備調査 ンステム計画 知識収集 ンステム設計 知識構造化 プログラム設計 知識ベース設計 プログラム作成 知識ベース構築 テスト、評価 実用化、運用

開発の主体が、①研究者、②コンサルタント、③ユーザ（企業の情報システム部門）によって、開発のライフサイクルに対する考え方が異なる。
研究開発するときは、プロトタイピングアプローチである。日本ではあまり行われていないが、コンサルタントが想定するライフサイクルは、初期にリリースしたバージョンを契約により、次々とバージョンアップいていくやり方を取る（付録Ａ．海外調査ｼﾑﾌﾚｯｸｽﾃｸﾉﾚｯｼﾞ社（Hayes-Roth博士）を参照）。
また、ユーザは従来のソフトウェア開発と同様に要求分析のアプローチから始まる。この様な現状と各種のエキスパートンステムのライフサイクルが提案されているが、これはその立場を考えると各々妥当なものである。
我々は、プロトタイピングに基づくフェーズドアプローチによるライフサイクルモデルを構成することとした。すなわち、開発当初のおいてシステムとしての本質的な要求を定義し、システムを徐々に成長させていく。各フェーズで段階的に要求仕様を明確化していくアプローチを採用する。そして、各フェーズが終了する毎に、直前の段階で定義された要求仕様に対するＶ＆Ｖが実施され、次のフェーズへ向けての仕様が決定する。
この考え方に沿い、我々は、図2.5-1のようにライフサイクルを設定した。このライフサイクルは、日本のツールベンダー、メーカー、大規模ユーザの実情にあっている。

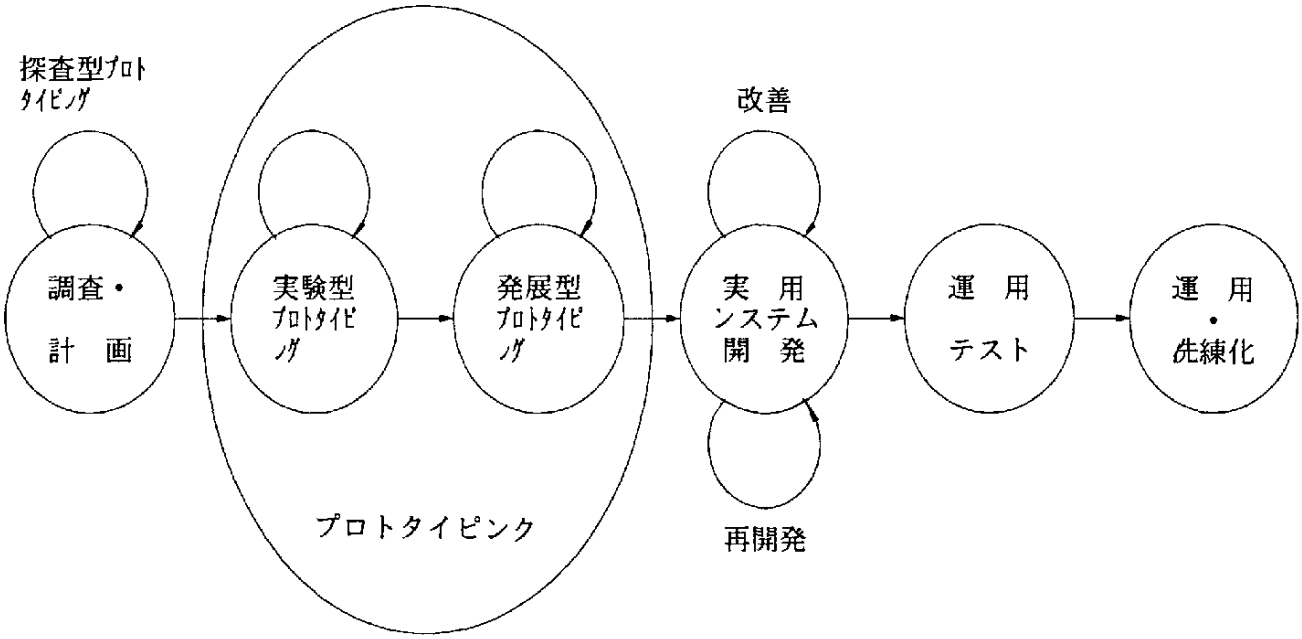


図２５－１　エキスパートンステム開発ライフサイクル

(1) 調査・計画フェーズ

このフェーズは従来のシステム開発における初期フェーズと同しで、要求仕様、システムの位置づけ（他のシステムとの関連性）、プロジェクト開発体制、開発スケジュールを明らかにする。この段階で重要なことは、企画提案書（第0版要求仕様書）を作成することである。企画提案書は、企画グループが作成する。即ち、利用部門のマネージャ、開発部門のマネージャ、運用部門のマネージャ、ドメインの専門家、エンドユーザのメンバーで構成される。

また、この段階で企画提案書を固めるために、探査型プロトタイピング（Exploratory prototyping）が行われる場合もある。以下は、この段階で行うべき作業内容である。

1 テーマの提起
2 予備調査
　現状の問題点／ニーズの調査分析
　システム化可能性の調査
　問題を解決する技術ポイントの調査
　対象範囲の検討
3 システム開発の計画
　チーム編成の検討
　運用イメージの検討
　システム構成の検討
　ツール・言語・ハードウェアの選択
　開発効果とコストの見積りの検討
4 企画提案書の作成
　システム化の目的と範囲
　プロジェクト要件
　　開発費、納期、要員計画
　実現したい機能
　導入効果
5 評価：開発可能性の検討

図2.5-2　調査計画フェーズの作業内容

(2) 実験型プロトタイピングフェーズ

このフェーズでは、実験的にプロトタイプを作成する。作成されるプロトタイプは、目的を重点的に絞り、知識量、性能、ヒューマンインタフェース機能等は実用レベルに達しなくてもよい。ここでは技術的に核になる知識・推論機能がともかく実現され、技術的な実現可能性の検証に重点がおかれる。このプロトタイプにより、実用化のめどをたてる事が重要である。設定された目的か達成できなかった場合には、再度挑戦することもある。

以下は、この段階で行うべき作業内容である。

1 要求仕様第Ⅰ版の作成
　システム化の目的と範囲（このフェーズにおける）
　プロジェクト要件
　機能要件
2 概念設計
　業務内容の詳細把握
　　業務の分析
　　知識の収集・分析
　　専門家への面接
　　事例の収集
　専門家モデルの設定
3 構造設計
　問題解決のメカニズムの設計
　　類型的タスクの検討
　　知識表現の検討
　　推論制御の検討
　データ構造の設計
4 知識プログラミング
　ルール、対象モデルの属性、手続き等の記述
　核となる問題のインプリメント
　テスト
5 評価（ＶＶＴ）：プロジェクト継続可否の検討
6 代替案の創出
7 開発体制の評価
8 テスト手法の検討

図2 5-3　実験型プロトタイピングフェース

(3) 発展型プロトタイピングフェーズ

このフェーズでは、規模的には小規模でもシステムの機能が見えるレベルのプロトタイプを作成する。このフェーズの最終段階では、プロトタイプを関係者に対しデモを行い、実用システムの開発の可否が判断できるレベルに達しているものとする。また、エンドユーザ、開発者、専門家のそれぞれが、このフェーズ後半にはシステムイメージが明確になり実用システムの要求仕様を詳細に記述できるレベルになるよう、プロトタイプの作成を行う。なお、このフェーズの途中で、必要の都度デザインレビューが行われる。

以下は、この段階で行うべき作業内容である。

1 要求仕様第Ⅱ版の作成
　システム化の目的と範囲（このフェーズにおける）
　プロジェクト要件
　機能要件
　性能・品質要件
　導入効果
2 構造設計
　問題解決過程のメカニズムの設計
　データ構造の設計
3 知識プログラミング
　問題解決過程の詳細化
　　ルールの設定
　　手続きの機能整理
　対象モデルの詳細化
　　オブジェクトの定義
　　属性定義の修正
　インプリメント
　テスト
4 評価（ＶＶＴ）：実用システムの開発可否の検討

図2 5-4　発展型プロトタイピングフェーズの作業内容

(4) 実用システム開発フェーズ

プロトタイプが評価用またはデモ用であるのに対して、実用システムとは実際に業務内に組み込まれて利用されるシステムであると定義する。このフェーズでは従来のシステム開発と基本的にはなんら変わりはない。ただ、このフェーズにいたるまでにプロトタイプを作成しているため、その財産（仕様・知識・技術・プログラム等）を有効に利用できる。しかし、必ずしもプロトタイプシステムを改良して実用システムにして行くべきであるというわけではない。実用システムの仕様に合わせて、言語・ツール、推論機構等を見直して、再構築するアプローチをとる必要もある。以下は、この段階で行うべき作業内容である。

1 実用版要求仕様の作成
　システム化の目的と範囲
　プロジェクト要件
　　開発費、納期、要員
　機能要件
　　要求される機能、入力データの範囲、出力結果
　ユーザ要件
　　導入効果、ユーザインタフェース
　性能・品質要件
　　問題解決能力、応答時間、正確性、保守性、
　　使用容易性
　環境要件
　　資源の制約、ハードウェアの制約
2 改善のとき
　問題解決能力の強化
　ヒューマンインタフェースの強化
　性能の向上
　運用環境への移行
　　保守用ツールの検討
3 再開発のとき
　チーム編成の検討
　ツール・言語・ハードウェアの選択
　構造設計
　　問題解決のメカニズムの設計
　　知識表現の検討
　　推論制御の検討
　　データ構造の設計
　知識プログラミング
　　ルール、対象モデルの属性、手続き等の記述
　　核となる問題のインプリメント
　　テスト
　運用環境への移行
　　保守用ツールの検討
4 評価（ＶＶＴ）：実用版要求仕様の達成度評価

図2 5-5　実用システム開発フェーズの作業内容

(5) 運用テストフェーズ

このフェーズでは、システムをフィールドでテストする。テストデータによる検証、従来システムとの並行ランによって、システムを評価し運用に耐えられるレベルまで修正する。テストは長期間に渡る場合が多い。以下は、この段階で行うべき作業内容である。

1 テスト環境の整備
2 運用テストの実施・システムの修正
3 保守用ツール、ユーティリテイの作成
4 運用方法の説明
　保守用ドキュメントの整備
　　保守・拡張方法
　　運用上の注意事項（ユーザ向け、開発者向け）
5 評価（ＶＶＴ）：デリバリ可否の検討

図2.5-6　運用テストフェーズの作業内容

(6) 運用・洗練化フェーズ

このフェーズは、システムを実運用していきながら、システムの問題解決能力を一層向上させていくフェーズである。システムは開発グループから運用グループに引き渡され、必要な保守（能力向上）はユーザと運用担当者グループの協力で行われるものとする。なお、変更に対する最終責任者は運用担当マネージャである。以下は、この段階で行うべき作業内容である。

1 運用
2 洗練化
　知識ベースの拡張・修正
3 変更要求仕様書
　保守性
　変更の容易性
　拡張性
4 評価（ＶＶＴ）：利用者の満足度

図2 5-7　運用・洗練化フェーズの作業内容

2 5.3 開発フェーズと評価対象との関係

前述したが、我々の設定したライフサイクルでは、各開発フェーズで段階的に要求仕様が明確化されていくアプローチを採用している。そして、各開発フェーズが終了する毎に、その段階の最初に定義した要求仕様、および各種トキュメントとソフトウェア（調査計画の段階では、原則としてないが）てＶ＆Ｖが実施され、次のフェーズに向けての仕様が決定する。

この考え方から、評価対象は、各開発フェーズで作成したソフトウェアとトキュメントとなる。作成するトキュメントは、図2 5-8 のような体系になっている（詳細は、2.6.2節を参照）。原則として、必要なドキュメントの種類とその内容についてはすべて整備してある（これを、「フルスケールトキュメント」と呼ぶ）。このうち、要求仕様書と知識分析書の２つのドキュメントは、開発担当者では不明確な情報を整理したものであるから必須なドキュメントと位置づける（ただし開発担当者が、エントユーザや知識提供の専門家である場合もあるが）。しかし、ドキュメント作成が現状ではあまり実施されていないことを考慮すると、この２つのドキュメント以外は必要に応じて使うことが現実的である。

要求仕様書は、各開発フェーズの最初に書かれるものとする。また評価作業は、原則として各フェーズの最後に行う。図2 5-8 は各フェーズでの評価対象を表している。

開発フェーズ	区分	調査・計画（探査型プロトタイピング）	プロトタイピング		実用システム開発	運用テスト	運用・洗練化
			実験型プロトタイピング	発展型プロトタイピング	改善 再開発		
評価対象	ソフトウェア	—	第Ⅰ版プロトタイプ	第Ⅱ版プロトタイプ	実用システム	実用システム	デリバリシステム
	トキュメント	企画提案書（第0版要求仕様書）	第Ⅰ版要求仕様書	第Ⅱ版要求仕様書	実用版要求仕様書		変更要求仕様書
			第Ⅰ版知識分析書	第Ⅱ版知識分析書	実用版知識分析書		
		調査報告書	第Ⅰ版問題解決分析書 第Ⅰ版システム設計書 第Ⅰ版テスト文書	第Ⅱ版問題解決分析書 第Ⅱ版システム設計書 第Ⅱ版テスト文書	実用版問題解決分析書 実用版システム設計書 実用版テスト文書 運用マニュアル 保守マニュアル	運用テスト版テスト文書	運用実績書 保守報告書 改訂版ドキュメント一式

図 2 5-8 開発フェーズと成果物・評価対象

注）□は、必須トキュメントである。他は必要に応して使うことも可能である。

2 5.4 評価者について

エキスパートシステムを評価する人は、複数存在する。それぞれの立場は以下の通りである。

(1)マネージャ　：次のフェーズへ進む決断を行う人である。
(2)開発担当者　：開発グループの中でも、主として段階毎のデザインレビューや、ウォークスルーを行う人達である。
(3)専門家　　：知識提供者である。
(4)第３者専門家：当該分野の専門家で、知識提供者とは異なる人である。
(5)ユーザ　　：専門家支援システムでは、ユーザは専門家で、専門家の代行システムでは、ユーザは非専門家である。
(6)運用担当者　：実用システムの管理・運用を行い、システムの変更の責任を担う人達である。
(7)企画グループ：企画グループは、システム開発に先立ち、システムの調査・計画を行うグループであり、下記のメンバーから構成されている。

- 利用部門のマネージャ
- 開発部門のマネージャ
- 運用部門のマネージャ
- エントユーザ
- ドメインの専門家

2.6 ドキュメント作成に対する基本的な考え方

2 6.1 ドキュメントは何処まで必要か

ドキュメントには、「ミニマムトキュメント」から「フルスケールトキュメント」に至るまで各種の作成方法がある。例えば、米国国防総省では高品質のシステムの開発をするため、ソフトウェア開発標準DOD-STD-2167A が定められている［DOD 1985］。そのドキュメント体系は表2 6-1 で述べてあるが、このうちエキスパートシステムのＶ＆Ｖを行う上で有用なドキュメントとして、［Wright 1990］によると、

- システム仕様(SSS)
- ｼｽﾃﾑ/ｾｸﾞﾒﾝﾄ設計ﾄﾞｷｭﾒﾝﾄ(SSDD)
- ｿﾌﾄｳｪｱ開発計画(SDP)
- ｿﾌﾄｳｪｱ要求仕様(SRS)
- ｲﾝﾀﾌｪｰｽ要求仕様(IRS)

の５種類を上げている。

このドキュメント標準は、ウォーターフォール型、ラピッドプロトタイピング型、スパイラル型のモデルに合い、詳細かつ緻密に記述する形式である。この種のドキュメントを「フルスケールドキュメント」と呼ぶ。

フルスケールドキュメントに対して、エキスパートシステムの場合はミニマムドキュメントの方が現実的である。ミニマムドキュメントとは、「要求仕様書」と「知識分析書」である。要求仕様書は、ユーザからの要求をとりまとめたもの、知識分析書は専門家から知識獲得した結果を取りまとめたものであり、知識エンジニア（ＫＥ）がユーザでも専門家でもないとしたならば、少なくともこの２種類の外部仕様のドキュメントがなければ、ンステムの構築する際のより所がなくなる。

またミニマムドキュメントでは、システムを如何に構築したのかを示す設計書に相当するものがないことになる。これはある意味では、知識ベースのソースコードの理解性が優れていることで代替しているという考え方もあるからである。エキスパートシステムのツールは一般に高機能であり、直接知識を入力することが苦にならないため、あまり手書きのトキュメントは必要ではないと考えられている。また今後ツールの発展にともなって、ドキュメントからソースコートの生成、またはソースコードからドキュメントの生成を期待すへきであるという意見もある。

以上、２つの立場はあるが、我々は、その両方の立場をサポートする方針とした。すなわち、フルスケールドキュメントを用意し、適宜その中から選択することが出来るものとする。

表2.6-1　DOD-STD-2167Aドキュメントの内容

開発ベース	ドキュメントの種類	レビュー監査
システム要求分析	• 基本システム仕様	システム要求レビュー
システム設計	• システム仕様(SSS) ★ • ｼｽﾃﾑ/ｾｸﾞﾒﾝﾄ設計ﾄﾞｷｭﾒﾝﾄ(SSDD)★ • 基本ｿﾌﾄｳｪｱ要求仕様 • 基本ｲﾝﾀﾌｪｰｽ要求仕様 • ｿﾌﾄｳｪｱ開発計画(SDP) ★	システム設計レビュー
ｿﾌﾄｳｪｱ要求分析	• ｿﾌﾄｳｪｱ要求仕様(SRS) ★ • ｲﾝﾀﾌｪｰｽ要求仕様(IRS) ★	ｿﾌﾄｳｪｱ仕様ﾚﾋﾞｭｰ
基本設計	• ｿﾌﾄｳｪｱ設計ﾄﾞｷｭﾒﾝﾄ(基本設計) • ｿﾌﾄｳｪｱテスト計画 • 基本ｲﾝﾀﾌｪｰｽ設計ﾄﾞｷｭﾒﾝﾄ	基本設計レビュー
詳細設計	• ｿﾌﾄｳｪｱ設計ﾄﾞｷｭﾒﾝﾄ(詳細設計) • ｿﾌﾄｳｪｱテスト記述 • ｲﾝﾀﾌｪｰｽ設計ﾄﾞｷｭﾒﾝﾄ	詳細設計レビュー
ｺｰﾃﾞｨﾝｸﾞとテスト	• ｿｰｽｺｰﾄﾞﾘｽﾃｨﾝｸﾞ • ソースコート	
CSC統合化とﾃｽﾄ	• ｿﾌﾄｳｪｱテスト記述(記述)	ﾃｽﾄ準備状況レビュー
CSCIテスト	• ソースコートの修正 • ｿﾌﾄｳｪｱテスト報告書 • 操作・支援ﾄﾞｷｭﾒﾝﾄ • 版記述ﾄﾞｷｭﾒﾝﾄ • ｿﾌﾄｳｪｱ製品仕様	機能・物理構成監査

注）　ＣＳＣ：ｺﾝﾋﾟｭｰﾀｿﾌﾄｳｪｱ構成要素、他のＣＳＣ及び単位プログラムから成る。
ＣＳＣＩ：ｺﾝﾋﾟｭｰﾀｿﾌﾄｳｪｱ構成項目、ＣＳＣと単位プログラムから成る。
★印　：ｴｷｽﾊﾟｰﾄｼｽﾃﾑのV&Vを行う上で有用なﾄﾞｷｭﾒﾝﾄ［Wright　1990］

2.6.2 エキスパートシステム開発／保守用ドキュメントの仕様

本プロジェクトでは、フルスケールトキュメントの用意をしておく。設定した各開発フェーズごとに開発用トキュメントあるいは保守用ドキュメントが作成されるものとする。従来、エキスパートシステムの開発においてはトキュメントが満足に作成されているシステムは数少ないが、今後実用化を目指すンステムにおいてはドキュメントは重要視される。ドキュメントは、レビューして確認することが出来るばかりでなく、大規模システム化を図るとき担当者間でのコミュニケーションを行うためにも必要である。エキスパートシステムにおけるトキュメントは知識ベースに関するものが中心となり、その保守に際してはきわめて重要な役割を果たす。

以下各ドキュメントの内容について述べる。

(1) 企画提案書（第0版要求仕様書）
企画提案書には、下記の内容から適時選択され、盛り込まれる。

1.システム化の必要性、目的（目標）
2.システム化の最終イメージ
3.システム化の範囲
4.実現すべきシステム機能（必須、オプション別）
5 性能・品質要件
- 目標とする問題解決能力
- アベイラビリティ、保守性、使用容易性など

6 目標とするシステムのコスト／パフォーマンス要件
7 ユーザ要件（専門家／非専門家、特定／不特定、利用場所）
8 データ要件
9.統合化要件（例えば、データベース、既存ソフトウェア、通信）
10.セキュリティ対策要件
11.開発ハートウェア、実行ハードウェア要件
12 ツール・言語要件
13.開発方法要件
14 プロジェクト要件
- 開発体制
- 総開発費（システム開発の生産性目標）
- 開発スケジュール（主たるマイルストーン）

図2 6-1 企画提案書

(2) 調査分析書

調査分析書には、下記の内容等が記述される。

- 現状の問題点
- あるべき姿
- 問題を解決するための技術ポイント
- 世の中の動向

図26-2 調査分析書

(3) 要求仕様書

要求仕様書には、各版の発展レベルに応じて、下記の内容から適時選択され、盛り込まれる。

1. システム化の必要性、目的（目標）
2. システム化の最終イメージ
3. システム化の範囲
4. 実現すべきシステム機能（必須、オプション別）
5. 性能・品質要件
   - 目標とする問題解決能力
   - 応答時間、正確性、保守性、使用容易性など
6. 目標とするシステムのコスト／パフォーマンス要件
7. ユーザ要件（専門家／非専門家、特定／不特定、利用場所）
8. データ要件
9. 統合化要件（例えば、データベース、既存ソフトウェア、通信）
10. セキュリティ対策要件
11. 開発ハードウェア、実行ハードウェア要件
12. ツール・言語要件
13. 開発方法要件
14. プロジェクト要件
    - 開発体制
    - 総開発費（システム開発の生産性目標）
    - 開発スケジュール（主たるマイルストーン）

図26-3 要求仕様書

(4) 知識分析書

知識分析書には、下記のように知識源から収集した知識が整理・分析され、記述される。

1 知識および知識源の同定
- 存在する知識源
- 存在する知識の形態
- 知識源の信ぴょう性
- 知識の複雑性
- 知識の定式化／標準化の程度
- 知識の変化する可能性
- 機密性
- 知識獲得の可能性（アクセス可能性）

2 知識の獲得
- 文献からの収集知識（幅広い知識）
- 専門家からの収集知識（インタビューメモ）
- 観察からの収集知識（具体的な知識）
- 事例からの収集知識（不足する知識の発見、知識のテスト）

3.応用分野の用語の定義

4 知識の分析
- 知識関連図（構成図）
- 知識一覧表（知識の分類）
- 知識項目名称；型（制約、対象物やその属性、制御知識、ｱｸﾃｨﾋﾞﾃｨ）
  ；備考

図2 6-4 知識分析書

(5) 問題解決分析書

問題解決分析書には、下記のように専門家が実際に問題をどうやって解いているかを整理・分析して、記述される。

1.解の分析
- 解の内容
- 解の形式（提示媒体、提示形式（文章、表、グラフ、図形）、説明機能による説明文）

2.制約条件の分析
- 強い制約
- 弱い制約
- 制約間のトレードオフ

3.問題解決過程の分析
- 問題解決モデル（専門家モデル）
- 問題の分割（部分問題化）
- 問題解決パラダイム（ジェネリックタスク）
- 知識の表現形式
- 対象モデルの表現形式（データ構造）
- 推論方法
- 競合解消戦略
- 不確実性情報に対する処理方式
- 学習の方式

4 ユーザモデルの分析
- 専門知識のレベル
- 推論制御の主導権（対話／非対話）
- ユーザインタフェース
- 利用場所（オフィス、工場）

5 機能の分析
- 機能階層図
- 機能項目名称；機能概要；実現形態(ルール、メソット／関数／手続き)；用途（問題解決、入出力）
- 機能オブジェクト関連図

図2 6-5 問題解決分析書

(6) システム設計書

システム設計書には、下記のようにコンピュータでどう実現するかの詳細が記述される。

1. システム処理の全体構想
   - システムの概念フロー図とその説明
   - オペレーションの全体フロー図とその説明
   - ユーザとの会話シナリオ
2. ワーキングメモリの設計
   - ワーキングメモリ定義
   - 変数名; データタイプ; 初期値
3. フレーム／オブジェクトの設計
   - フレーム構成図
   - フレーム一覧表
     - フレーム名称; 概要; 主なスロット; 主なメソッド; 備考
   - フレーム定義
     - フレーム名; super-class
     - スロット名; データタイプ;初期値;値域; デモン (asked、 changed)
   - メソッド名
4. ルールの設計
   - ルール構成図
   - ルール一覧表
     - ルール名称; 概要; 戦略; 入力; 出力; 備考
   - ルールの実行順序関連図
     - ルール名称; 先行ルール (群) 名称; メタルール; 備考
   - ルール定義
     - ルール名
     - 目　的
     - 所属ルール群名; 確信度
     - 条件部 (IF) ; 実行部 (THEN)
     - 条件部 (IF) の説明; 実行部 (THEN) の説明
     - 関係オブジェクト (参照するスロット、 変更するスロット)
     - ルールの出所 (出典) 箇所
5. メソッド／関数／手続きの設計
   - メソッド関連 (構成) 図
   - メソッド一覧表
     - メソッド名称; 種別 (処理、 入力、 出力) ; 処理概要
   - メソッド／関数／手続き定義
     - メソッド名; 呼出し側フレーム名
     - 呼出し系列; 返す値; 使用外部メソッド (関数) ; アクセスするフレーム&スロット; 内部変数
     - 処理概要 (入力、 処理、 出力)
     - 詳細処理手続き (ロジック)
     - 補足説明

- 使用例

6.オペレーションの設計
- オペレーション一覧表
  - オペレーション；画面内容；処理内容；知識ベース

7.画面の設計
- 画面遷移図
- 画面一覧表
  - オペレーション；画面内容；処理内容；知識ベース
- 入力データ
- 表示（出力）データ
- 画面定義
  - 画面名；レイアウト；入出力項目

図2.6-6　システム設計書

(7) テスト文書

テスト文書には、テストデータの記述と実際にテストした結果を記述する。

1.テストケース番号
2 ケース作成者；日付
　テスト担当者；日付
　チェック者；日付
3.テストケースの記述
　期待する出力
　結果
4.テストによってカバレージしたルール

図2.6-7　テスト文書

(8) 運用マニュアル

システムを運用する際に、運用担当者をサポートするマニュアルである。

1 システムのインストール方法
2.システムの操作方法
3.例外、異常処理に対する対策

図2.6-8　運用マニュアル

(9) 保守マニュアル

システムを保守する際に、保守の担当者をサポートするためのマニュアルである。このマニュアルは知識分析書、問題解決分析書、システム設計書から保守に必要な部分を抽出して取りまとめてある。

1 オブジェクトとルールの関連ツリー 2 ルール間の関連ツリー 3 オブジェクト継承関係表

図2 6-9 保守マニュアル

(10) 運用実績書

システムを実際に運用した実績を取りまとめたものである。

1 運用日、問題点、解決方法、処置担当者

図2 6-10 運用実績書

(11) 保守報告書

保守を行った結果を取りまとめたものである。

1 保守日、保守事項、保守担当者

図2.6-11 保守報告書

2.7 ガイドラインの使い方

2.7.1 ガイドラインの想定利用者と利用時期について

本ガイドラインの利用者は、エキスパートシステムの開発にたずさわる開発担当者もしくはプロジェクト管理担当者を想定した。このような担当者レベルの技術者や管理者は、より品質の高いシステムを実現する義務をおっており、エキスパートシステムのV&Vの概念を明確に理解することが重要である。

さらに、この想定利用者は、（必ずしもエキスパートンステムに限らない）ソフトウェア開発プロジェクトの経験を持ち、かつ、人工知能の基礎概念についてひととおりの知識を持つことを仮定している。したがって、本ガイドラインでは、ソフトウェア開発プロジェクトに対する一般的な考察や、人工知能技術に対する解説をできるだけ簡潔な範囲にとどめている。これらについて、より詳しい情報が必要な場合は、適宜参考文献を参照されたい。

また、評価項目の利用時期としては、調査・計画の段階から、ライフサイクルに従い、実運用システムが完成し、運用・保守にはいるまで、各フェーズごとに利用できるものを目指した（詳細は、3.2ライフサイクル・評価者別評価項目を参照のこと）。チェックリストに示した諸項目は、人手によるレビュー時、ツールを用いたテスト時のいずれの場合にも適用できるものである。表2 7-1に、開発フェーズごとの評価の時点を述べる。

表2 7-1　評価の時点

開発フェーズ	開発作業	評価
調査・計画	企画提案書作成 （探査型プロトタイピング）	デザインレビュー
実験型プロトタイピング	第Ⅰ版要求仕様書作成 設計 実験型プロトタイプ作成	デザインレビュー 評価（VV&T）
発展型プロトタイピング	第Ⅱ版要求仕様書作成 設計 発展型プロトタイプ作成	デザインレビュー 評価（VV&T）
実用システム開発	実用版要求仕様書作成 改善または再開発 実用システム作成	デザインレビュー 評価（VV&T）
運用テスト	実用システムの運用テスト	評価（VV&T）
運用・洗練化	実用システムの運用と洗練化	評価（VV&T）

2.7.2 本ガイドラインの利用の仕方

本ガイドラインを利用する際の基本的な手順を図2.7-1 に示す。本ガイトラインで提供している点検項目・測定項目は非常に広い範囲の問題をカバーし得る包括的なものである。従って、これを実際に利用する場合には、エキスパートシステム開発プロジェクトを開始する前に、自らの問題に合わせて適切な評価項目を取捨選択し、対象システムの特性に合致した、評価方法を準備しておくことを薦めたい。それには、２７３節で述べる方法が参考となろう。

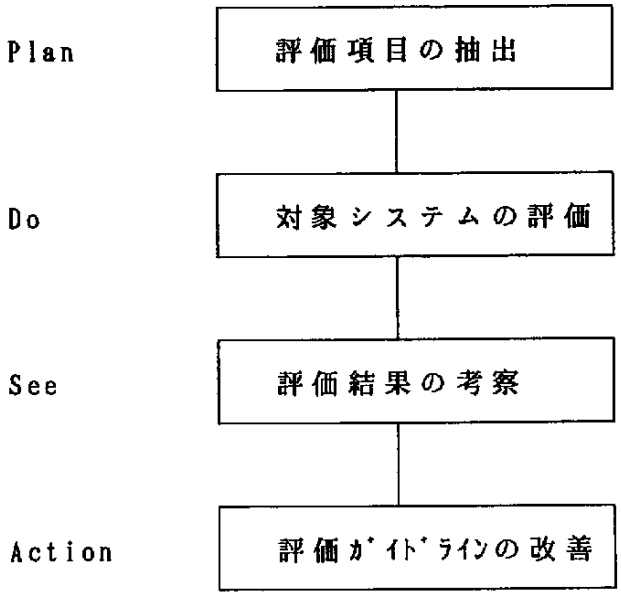


図２７-1　利用の手順

(1) 評価項目の抽出

評価を全てのチェック項目にわたって行うことは、現実的には時間的、コスト的に不可能な場合が多い。従って、特に効果が高いと思われる視点を予め検討して、重点管理をすることが大切である。評価項目の抽出に当たっては、利用目的に応じてチェックリストを複数の視点で判断できるような分類方法を３１節で示す。ただし、デザインレビューを行う時点においては、検討の漏れがないことを確かめる上でも、ここで提供した評価項目全てについて評価することは価値があると思われる。

また、評価項目のチューニングを行う場合は、以下で紹介する「品質機能展開ＱＦＤ(Quality Function Deployment)」、「リスクを評価する手法」、「ベンチマーク基準」等の考え方が参考となろう。

(2) 対象システムの評価

デザインレビューの実施に当たっては、上で述べたように、本ガイトラインで提供している点検項目・測定項目にすべてに基づいて評価を行うことが望ましい。評価を行う際、専用のチェックリスト（点検項目・測定項目に対しての実現の度合、コメント等を含む）を作成して評価を行うと後の整理がしやすいと考えられる。また評価を行った結果は、数量化しておくことが望ましい。またはデモなどを実演して評価を行うときは、プロトコル分析等の手法を利用してとりまとめ、

チェックリストに反映することが考えられる。

本ガイドラインでは、評価結果を数量化する手段については、基準を設けていない。これは、現在のところ提案されている手法が、どれもアトホックな手順に基づいており、ガイトラインとして一定の方法を提案するのは時期尚早と考えたためである。

ただし、総合的な評価結果が必要な場合は、例えば適当な方法で、各項目の重要度に重みをつけてその重み付け平均で評価することが一般的であろう。ソフトウェア品質管理の上では、階層化意思決定法（ＡＨＰ手法）がその際に簡便かつ有用な手法として推奨されている［Yoshizawa 1988］。

また、評価結果をより分かりやすくするため、蜘型グラフや表などを利用して整理する方法も［Yoshizawa 1988］では推奨されている。

(3) 評価結果の考察

評価結果に基づいて、システム開発の次のフェーズへの移行の可否、プロジェクト継続の可否が決定される。その際の留意点は以下の通りてある。

①デザインレビューのやり方に慣れること

実施する際の時間的な効率を検討しておくこと。

②形式的なチェックに流れないようにすること

点検項目の数の多さに流されないようにすること。このため点検項目・測定項目について評価した結果、例えば各種数値データや未達成項目、問題点と原因・対策などを記録する管理表を用意しておくことも考えられる。

③チェックの業務分担を明確にしておくこと

「３２ライフサイクル・評価者別評価項目」を適用して、各担当者の役割を決めておくこと。目標値に達しなかった場合やアフターケアは、マネージャの責任と権限で柔軟に対処できるようにしておくこと。

④評価には慎重に当たること

- 評価の判断が出来にくい場合、客観的に判断できるその道の専門家、またはユーザに委ねること。
- 特定の点検項目が満足されていなければ、例え他の項目がすへて満足されていても不合格と言うことも考えておく必要がある。

⑤調査・計画の段階から適用すること

全工程に渡って、初期の段階から適用していく体制を作ることが重要である。

(4) 評価ガイトラインの改善

４章　本ガイトラインの適用と評価を参照するとわかるように、１つのエキスパートシステム開発に本ガイドラインを適用して評価を行ったとしても、それによって、一般的な評価手順が確定するわけてはない。従って、品質管理運動てしばしばなされるように、ガイトラインの改善活動を継続していくことは重要である。この際には、ソフトウェア工学における品質管理活動を参考にすることができる（例えば、［Deutsch 1990］を参照）。

2.7.3 本ガイドラインを更に有効に利活用するための方法

以下に、本ガイドラインを更に有効に利活用するための方法（具体例）の概要について説明する。

- 品質機能展開手法　ＱＦＤ(Quality Function Deployment)
- リスクを評価する手法
- ベンチマーク基準
- チェックリストの作成
- 重み付け平均の方法
- ＡＨＰ(Analytic Hierarchy Process)

(1) 品質機能展開手法　ＱＦＤ(Quality Function Deployment)　[IPA 1989]

品質機能展開は、顧客の要求品質を分析把握し、それを代用特性に変換して製品の設計品質を定め、これを機能部品やユニットの品質、更に個々の構成部品の品質から工法や工程の要素に至るまで、それらの間の関連を明確に展開して、顧客の真の要求品質を完成品にまで伝播して品質を作り込んでいく手法である。

米Ford Motor社では、製品デザインの最も重要な過程、つまりユーザは何を求めているのか、その実現手法は何か、コストはどの程度なのか等、設計に反映される各種の要因とその相関関係を品質機能展開の方法により分析を支援するツールを作成した[斎野 1990]。図2.7-2に品質機能展開の各表の概略作成フローを示す。

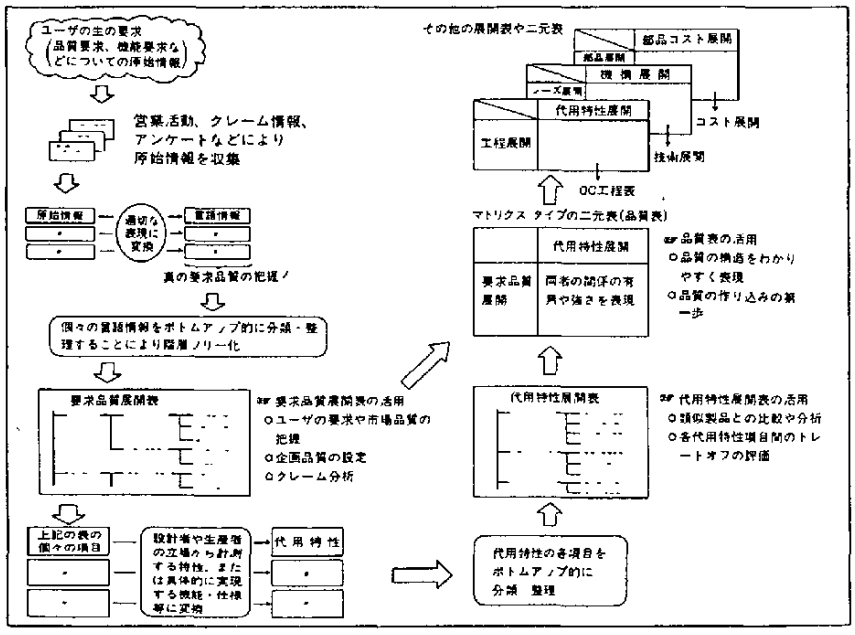


図2.7-2　品質機能展開の各表の概略作成フロー

(2) リスクを評価する手法　[Mockler 1990]

Mockler は、知識ベースシステムを用いて種々の知識ベースシステム開発計画に内在するリスクの評価方法(Project Risk Evaluator)を開発した。

Project Risk Evaluatorシステムは、技術的要因、組織的要因及びプロジェクト資源の十分性という３つの領域を評価する。図2.7-3に技術的な要因の依存間系図を示す。図2.7-4 に初期プロトタイプにおける、必要な専門家についての意思決定チャートを示す。

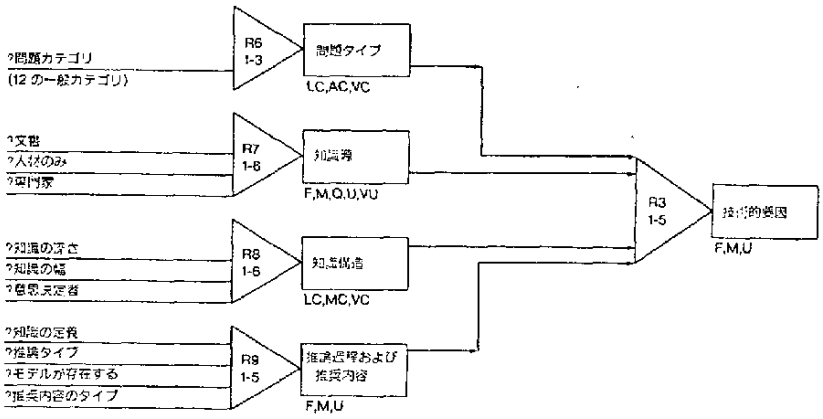


図2.7-3　技術的な要因の依存間系図（日経ＡＩ 1990夏号p.130）

初期バージョン

ルール セット19.	1	2	3	4	5	6	7
問題解決ユニット	FW	FW	M/MY	M/MY	M/MY	FW/M	FW/M
状況タイプ			VC/	VC/		LC/AC	LC/AC
知識源			U/Q/	U/Q/		F/M	F/M
知識構造		VC/	VC/	VC/		LC/MC	LC/MC
推論過程－推論内容		U	U	U		F/M	F/M
専門家タイプ					EE		
専門家タイプ×2					EEx2		
then 必要な専門家 (experts-required)=							
開発に一人	X						
テストにもう一人	X						
高い認知的技能力		X	X				
高い対人技能力				X			
専門家の人数					EEx2		
各専門領域から二人ずつ					X		
平均的な対人技能力						X	
平均的な認知的技能力							X

要約バージョン

ルール・セット22.	1	2	3	4	5
問題解決ユニット	FW	FW	M/MY	M/MY	FW/M
状況タイプ			VC/		LC/AC
知識源			U/Q/		F/M
知識構造		VC/	VC/		LC/MC
推論過程－推論内容		U	U		F/M
専門家タイプ				EE	
専門家タイプ×2				EEx2	
then 必要な専門家 (experts-required)=					
開発に一人	X				
テストにもう一人	X				
高い認知的技能力		X	X		
高い対人技能力			X		
専門家の人数				EEx2	
各専門領域から二人ずつ				X	
平均的な対人技能力					X
平均的な認知的技能力					X

FW＝少数　MY＝多数　VC＝非常に複雑　LC＝比較的複雑さが低い　AC＝平均的複雑度　MC＝中間的複雑度
F ＝有利　M ＝中間　Q ＝疑問　UN＝不利　EE＝技術
/ ＝あるいは　• ＝どの回答でもよい

図2.7-4　初期プロトタイプにおける、必要な専門家についての意思決定チャート（日経ＡＩ 1990夏号p.131）

(3) ベンチマーク基準　[Hayes-Roth 1989]

Hayes-Rothは、（まだ概念的ではあるが）知識ベースシステム向けの新しいベンチマーク手法を提案している。エンジニアリングの質に係わる内面的な(intrinsic)基準と、仕事の生産性に係わる外面的な(extrinsic)基準に大きく分けて、1つの評価方法として提案している。

よいベンチマークのために基準とは、①精密(Precise)であること、②様々な問題の代表的(Representative)な例を組み合わせたもの、③異なった状況で同じベンチマークを行うときは、全く同じ(Replicatable)結果を得るべきである、④それぞれのベンチマークは分割可能な(Segmentable)事例の集まりであること、⑤ベンチマークは時間と共に拡張(Extensible)でき、⑥専門家や研究者が快く賛成できるようにベンチマークは説得力(Persuasive)を持つべき、⑦できるたけ安価(Inexpensive)に実行できるものであるべきである。

精密　(Precise)
代表的(Representative)
再現性(Replicatable)
分割性(Segmentable)
拡張性(Extensible)
説得性(Persuasive)
安価　(Inexpensive)

図2 7-5　よいベンチマークの基準

努力すれば測定可能なベンチマーク基準として、図2 7-6の項目を上げている。

1 知識増加のコストと影響
2 無関係な要因に対する感度
3 結果の理解可能性
4 知識利用の徹底性
5 選択の適切さ
6 単位知識当りのコスト
7 容量と性能の限界
8 資源の限界に対する応答の適用性
9 知識削減の容易さ
10 ﾗﾝﾀﾞﾏｲｽﾞされた状態での測定基準の感度
11 知識のコート拡張性
12 ｺｰﾄﾞ化された知識の新しいﾀｽｸへの再利用性
13 知識管理ユーティリティ

図2.7-6　測定可能なベンチマーク基準

(4) チェックリストの作成

評価する際には、3章で述べる点検項目に対して、様式を作成して、容易に評価が出来るようにすることが必要である。

表2.7-2 チェックリスト(例)

点検項目	評価者	重要の度合	実現の度合	コメント

(項目の説明)

評価者　：評価担当者を明記する。

重要の度合：その点検項目が、いま対象としているシステムに対して関連している度合を記号や数値を用いて示す。

実現の度合：関連がある項目で、点検をした結果を記述する。記述は、関連の度合と同様である。

コメント　：点検した内容について、問題点、その原因と対策等、特に気がついた点をコメントする。

(5) 重み付け平均の方法　　[Slagle 1988]

Slagleは、対象とする問題はエキスパートシステム化が望ましいのかを評価・選択する方法として、重み付け平均の方法を採用した。項目の体系は、図2 7-7の通りである。この方法は、総合得点化することに意味のある場合に有効である。この方法は、項目の重要度に応じて、前もって重みを付けなければならない。なお、ここで提案されている重みの妥当性については、利用者がチェックを行う必要がある。

基本的な特徴
- ユーザとその管理者(3)
- タスク (10)
- 専門家 (11)

望ましい特徴
- ユーザとその管理者(5)
- タスク (7)
- 専門家 (4)

図2 7-7 評価項目の体系 ( )内は、項目数

総合得点を求める式は、次の通りである。

$$\frac{\Sigma\, weight_i * value_i}{\Sigma\, weight_i}$$

(6) ＡＨＰ手法に基づく評価　[Liebowitz 1986]

Liebowitz は、エキスパートシステムの評価基準を設定し、ＡＨＰ手法を用いてＲｅａｄシステム（NASAの衛星管理活動を支援するエキスパートシステム）に対して適用した。Ｒｅａｄで使用した評価基準は、表2 7-3 の通りである。

表2 7-3　評価基準

評価基準	定　　義
更新可能性	新しい入力に基づく変更を反映して出力を修正する能力。
使いやすさ	明確に理解でき、ユーザ親和性もあり、ｲﾝﾌﾟﾘﾒﾝﾃｰｼｮﾝが容易であること。
ハートウェア	実行に必要なｺﾝﾋﾟｭｰﾀ装置の移植性，使用可能性，ｱｸｾｽ可能性
費用面の効果	問題解決に必要なコストと得られる利益。
入出力内容	推論、ヘルプ機能、説明機能。
品　質	応答の正確さ、一貫性、完全性。
設計時間	問題の解決に必要な時間（人・年）。

《参考文献》

[AIｾﾝﾀ 1989]ICOT-JIPDEC AIｾﾝﾀ: 人工知能の技術と利用（ＡＩ白書），（財）日本情報処理開発協会，1989年12月．

[ANSI/IEEE 1988]東基衛: ANSI/IEEEソフトウェア規格集，日本規格協会，1988.

[Bobrow 1986]D.G.Bobrow, S.Mittal, M.J.Stefik: Expert Systems : Perils and Promise, Communications of the ACM, Vol.29, No.9, pp.840-894 ,1986.

[Buchanan 1983]B.Buchanan,et al.: Constructing an Expert System,Building Expert Systems, F.Hayes-Roth, D.Waterman, and D.Lenat, Addison-Wesley, pp.127-167, 1983.

[Deutsch 1982]M.S. Deutsch: Software Verification and Validation, Prentice-Hall,Inc., Englewood Cliffs, NJ, 1982.

[Deutsch 1990]M.S.Deutsch, R.R.Wilis（成田（訳））: ソフトウェア品質工学，日経ＢＰ，1989.

[EPRI 1988]J.A.Naser: Verification and Validation of Expert Systems for Nuclear Power Plant Application, EPRI Report, No.NP=5978, pp.116 , (August 1988).

[Geissman 1988]J.R.Geissman: Verification and Validation of Expert Systems,AI EXPERT, Vol.3, No.2, pp.26-33, (February 1988).

[Hayes-Roth 1987]Frederick Hayes-Roth: Expert System, Encyclopedia of Artficial Intelligence, Vol.1,2, John-Wiley, pp.287-298, 1987.

[Hayes-Roth 1989]Frederick Hayes-Roth: Towards Benchmarks for Knowledge Systems and Their Implications for Data Engineering, IEEE Transactions on Knowledge and Data Engineering, Vol.1, No.1, (March 1989).

[IPA 1989]情報処理振興事業協会技術センター: 品質機能展開による高品質ソフトウェアの開発手法「解説編」，コンピュータ・エージ社，1989年3月．

[Keller 1987]Keller: Expert System Technology: Development and Application, Yourdon Press, Englewood Cliffs, NJ, 1987.

[Kobayashi 1989]小林重信: Ⅶエキスパートシステム編，３章エキスパートシステム構築方法論，人工知能ハンドブック，人工知能学会，1990年1月．

[Liebowitz 1986]Jay Liebowitz: Useful Approach for Evaluating Expert Systems, Expert Systems, Vol.3, No.2, pp.86-96, (April 1986).

[Martin 1987]N.Martin: The Management of Expert System Development, tutorial presented at the 3rd Expert Systems in Government Conference, Washinton DC., 1987.

[Miller 1989]L.A.Miller: A Comprehensive Approach to the Verification and Validation of Knowledge-Based Systems, presented at the Workshop on Verification,Validation and Testing of Knowledge-Based Systems, IJCAI'89, 19 August, 1989.

[Mockler 1990]R.J.Mockler: Using Knowledge-based Systems for Estimating Risks Inherent in a Proposed KBS Project, Expert Systems, Vol.7, No.1, pp.27-41, (February 1990).
訳: 日経ＡＩ別冊１９９０夏号，pp.116-133.

[O'Keefe 1987]R.M.O'Keefe, O.Balci, E.P.Smith: Validating Expert System Performance, IEEE EXPERT, Vol.2, No.4, pp.81-90, (Winter 1987).

[O'Leary,D.E. 1987]D.E.O'Leary: Validating of Expert Systems - with Applications to Auditing and Accounting Expert Systems, Decision Sciences, Vol.18, No.3, pp.468-486, (Summer 1987).

[O'Leary,D.E. 1989]D.E.O'Leary, O'Keefe: Validating of Expert Systems, 11th IJCAI'89, Tutorial Program, 1989.

[O'Leary,T.J. 1990]T.J.O'Leary, M.Goul, K.E.Moffitt, A.E.Radwan: Validating Expert Systems, IEEE EXPERT, Vol.5, No.3, pp.51-58, (June 1990).

[Slagle 1988]J.R.Slagle, M.R.Wick: A Method for Evaluating Candidate, Ai Magazine, pp.44-53, (Winter 1988).

[Yoshizawa 1988]吉沢,東,片山: ソフトウェアの品質管理と生産技術,日本規格協会, 1988.

[Wright 1990]R.Glenn Wright: Expert System Verification and Validation, TUTORIAL #2,IEEE Managing Expert System Programs and Projects Conference, September 10-12, 1990.

[斎野 1990]斎野 亨: "知識武装化"を本格化させる米国先進企業,日経ＡＩ別冊, pp.22-35, 1990夏号.

[日立 1987]日立製作所: エキスパートシステム構築標準手順 ESGUIDE(第1版), SE-362, 1987.

[富士通 1988]富士通: エキスパートシステムの構築技法 ES/SDEM, テキスト, U0014L1H*2, 1988.

3.　エキスパートシステムの評価項目

3.　エキスパートシステムの評価項目

第３章は、本報告書ガイドラインの中心に当り、エキスパートシステムの評価項目について詳細な解説を行う。３１節は、全般的な評価項目とそれぞれの項目毎の点検項目と測定項目につき説明している。そして、３２～３５節では、ライフサイクル・評価者、システム利用形態、ドキュメント、問題のタイプの４つの観点からどのように分類・利用できるかを分析する。また、評価とはＶＶ＆Ｔであり、そのＴに当たるテスト手法について、３６節で体系的に説明する。

本章の目次構成は、以下の通りである。

３１ 評価項目とその解説
３２ ライフサイクル・評価者別評価項目
３３ ドキュメント別評価項目
3.4 利用形態別評価項目
3.5 問題のタイプ別評価項目
３６ 評価に用いられるテスト方法

3 1　評価項目とその解説

評価対象は2.3 2節で述べたように、①「知識ベースの妥当性評価」、②「問題解決・推論方式の妥当性評価」、③「結果の妥当性評価」、④「要求仕様に対する達成度評価」、⑤「知識ベースの保守性評価」、⑥「マンマシンインタフェースの妥当性評価」、及び⑦「システムの総合評価」から構成されている。

７つの評価対象をさらにブレークダウンした項目を「評価項目」と呼ぶ。評価項目は、表3 1-1 に示すように結果に関する評価（結果系、分類記号Ｐ）と要因に関する評価（要因系）とその他経済性・社会性に関する評価（分類記号Ｏ）の３つに大きく分類される。要因系は更に、評価方法そのものを評価（分類記号Ｅ）と方法論・技法・ツールの評価（分類記号Ｍ）に分けられる。

表3 1-1　評価項目の共通的な分類

結果系	P	プロダクトに関する評価
要因系	E	評価方法に関する評価
	M	開発／保守方法・技法・ツールに関する評価
その他	O	経済性、社会性に関する評価

以上の考え方に基づき、評価項目を分類・整理すると表3 1-2のようになる。

表3 1-2　評価項目

評価対象	分類	評価項目
1　知識ベースの妥当性評価	P	1-1　知識ベースの正当性評価 1-2　知識ベースの完備性評価 1-3　知識ベースの一貫性評価
	E	1-4　経験的知識を正当化する方法とその確認
	M	1-5　知識表現の選択と適切さの確認 1-6　知識獲得方法の選択と適切さの確認
2　問題解決・推論方式の妥当性評価	P	2-1　推論処理の妥当性評価 2-2　推論時間の妥当性評価
	M	2-3　推論手法の選択と適切さの確認 2-4　不確実情報に対する処理技法の適切さの確認 2-5　学習機能の適切さの確認
3．結果の妥当性評価	P	3-1　正解の評価 3-2　解の網羅性評価 3-3　解の最適性評価
	E	3-4　解の評価方法の適切さの確認
4．要求仕様に対する達成度評価	P	4-1　要求仕様内容に対する評価 4-2　要求仕様に対する達成度評価
	M	4-3　要求仕様設定手段の確認
5　知識ベースの保守性評価	P	5-1　変更箇所発見の容易性評価 5-2　知識ベースの理解性評価 5-3　変更箇所の局所性評価 5-4　知識ベースの統一性評価
	M	5-5　品質安定性の確認方法の確認 5-6　保守マニュアルの整備状況の確認 5-7　保守用ノールの整備状況の確認

表3 1-3　評価項目　（続き）

評価対象	分類	評価項目
6　ヒューマンインタフェースの妥当性評価	P	6-1　利用時のインタフェース評価 6-2　保守時のインタフェース評価
7　ンステムの総合評価	P	7-1　テーマ選定の妥当性評価 7-2　利用者の満足度評価 7-3　統合性の評価 7-4　成長可能性の評価 7-5　開発／保守用トキュメントの評価
	M	7-6　開発方法の適切さの確認 7-7　ツール・言語選択の適切さの確認
	O	7-8　システム開発の生産性評価 7-9　システムのコストパフォーマンス評価 7-10　セキュリティ対策の評価

以下に、各評価項目の解説を行う。該当する節を表3 1-4に示す。

表3.1-4　解説の該当節

3 1 1	知識ベースの妥当性評価
3 1 2	問題解決・推論方式の妥当性評価
3 1 3	結果の妥当性評価
3 1 4	要求仕様に対する達成度評価
3 1 5	知識ベースの保守性評価
3.1.6	ヒューマンインタフェースの妥当性評価
3 1 7	システムの総合評価

解説の内容は、評価対象については、

（1）定義

（2）目的と必要性

（3）説明　　（必要なときのみ）

評価項目については、

（1）定義

（2）目的と必要性

（3）説明

（4）点検項目(C)・測定項目(M)

である。

点検項目はC記号、測定項目はM記号を付加して番号付けをしてある。他の章、節で点検項目・測定項目が数多く流用されているが、その場合は同一の番号で対応を取ることにしてある。点検項目・測定項目は利用の便利さを考慮して、評価項目間での重複はあえて行うこととした。また、点検項目・測定項目の並び順序は、重要度の高い順番から並んでいる。

なお、3 2節以降で、ここに掲げた点検項目・測定項目以外に新たに作成している場合は、その番号の先頭が＊印である（点検項目のとき(*C1)など、測定項目のとき(*M1)など）。

3 1.1 知識ベースの妥当性評価

解説の該当節　　▨印
3.1.1 知識ベースの妥当性評価 3.1.1.1 知識ベースの正当性評価 3.1.1.2 知識ベースの完備性評価 3.1.1.3 知識ベースの一貫性評価 3.1.1.4 経験的知識を正当化する方法とその確認 3.1.1.5 知識表現の選択と適切さの確認 3.1.1.6 知識獲得方法の選択と適切さの確認
3.1 2 問題解決・推論方式の妥当性評価
3.1.3 結果の妥当性評価
3.1.4 要求仕様に対する達成度評価
3.1.5 知識ベースの保守性評価
3 1 6 ヒューマンインタフェースの妥当性評価
3.1.7 システムの総合評価

（１）定義

知識ベースに格納されている知識が、問題の対象領域において妥当であるのかを評価することである。

（２）目的と必要性

知識ベースの質が、結果の良否に最も直接影響を与える要因であるので、知識ベースの正当性、完備性、一貫性の評価を充分に行う必要がある。

《プロダクトに関する評価》

3.1 1 1　知識ベースの正当性評価

（1）定義

知識ベースに格納された知識の正しさを評価することである。

（2）目的と必要性

知識ベースに格納された知識自体が正確であることは、基本的な要件である。特に、経験的知識を取り扱わねばならないときには、その要求は強い。

（3）説明

専門知識は、教科書的知識と経験的知識に大きく分けられる。教科書的知識とは、一般的に実験や実証が可能、または公理や法則から導くことが出来る知識をいう。従って科学的な根拠、理論付けをすることが可能であり信頼性が高い。評価の観点からいうと、教科書的知識は正当性が裏付けられやすい。

一方、経験的知識は、一般に文書化されておらず、ときには当の専門家自身もその根拠に対し充分な説明が出来ない知識である。経験的知識の中には、旨い絞り方、無駄な思考を避ける知識、曖昧な情報を扱う知識などが含まれている。このため、効率よく複雑な問題を解決するのに役に立つ反面、科学的な根拠に乏しい、理論付けすることが難しいなどの欠点があり、慎重に取り扱わねばならない。

知識の正当性の評価では、その分野の第一人者といわれる専門家に依らざるを得ないことが多い。複数の専門家がいた場合、抽出出来る知識が異なることはもとより、同一対象に対する知識さえ異なることがある。このため、複数の場合、誰に決定権があるのかを明らかにしておかねばならない。

（4）点検項目(C)・測定項目(M)

(C01)［知識源の確認］［経験的知識の根拠］整理された知識は、知識源を遡りその出所を確認すると共に、その知識を用いることの利点、根拠を明らかにしておくこと。

(C02)［知識の型付け］獲得した知識は、型付け（ファクト、ルール、クラス、実体、関係、構造、行為、制約、資源）を行い、整理すること。
故障木の場合は、ノートとリンクの型付けを行う。

(C03)［知識の整理］用いる知識の重要度、詳細度、変化の可能性、使用頻度などの観点から、知識を整理すること。

(C04)［専門家によるレビュー］以上から、専門家（第3者専門家も含む）に概念レベル、意味レベルのレビューを行ってもらうこと。

(C05)［問題点の抽出・検討］その結果、専門家と開発者との間で知識に対する解釈の相違（多義性、曖昧性）に関して問題点を洗いだし、それを検討すること。

(C06)［知識の信憑性］［知識の適用範囲］特に、曖昧性、信頼性に欠ける知識については、他の専門家の意見を聞き、どの程度信憑性があるのか、又その知識の適用範囲についても検討すること。

(C07)［専門家間での合意］複数の専門家により抽出された知識が異なる場合、知識の解釈が異なる場合、専門家間で合意を得ること。

(C08)［ウォークスルー］エキスパートシステムが解決した過程と専門家が解決した過程とを比較検討して、どの知識が用いられているのかを逐一ウォークスルーすること。

(C09)［知識の追跡］あるケースにおいて、専門家の解決過程と結果が不明な場合があるが、その場合は専門家が言う知識に関して追跡することが重要であり、必ず明確にしておくこと。

(C10)［結果の視覚化］［解決過程の視覚化］エキスパートンステムの解決過程に専門家が納得行かない場合は、システム側で解決過程の説明をする、ルールとオブジェクトの関連ツリーを視覚化して表示する等の方法で、知識ベースの正当化を容易に検証できる機能を充分用意しておくこと。

(C11)［知識の制御］ある状況下において、活性化する知識（発火する知識）のタイミングや優先順位が適切であること。

(C12)［知識の正当性］その知識の正当性を検証するには、その知識が役に立たないことを実証できるかを確認すること。

3 1 1.2 知識ベースの完備性評価

（１）定義

対象とする問題を解くのに必要な知識が全て入っているのかどうかを評価することである。

（２）目的と必要性

必要な知識が欠けている場合、正しい結果が得られない可能性が高い。とくにキーとなる知識の欠落によって、目的とはまったく異なる結論となることすらある。まず対象とする問題の範囲を明確にし、それに対する必要充分な知識が格納されていることを確認することは極めて重要である。

（３）説明

知識ベースに対しては、プロトタイプ版では知識の完備性を完全に追求する必要はない。しかし、実用版をユーザに開放するときは、より完全な完備性を追求し、その評価が必要である。

また、エキスパートシステムにおいては、実運用段階に入っても更に知識ベースが更新されるため、その都度、知識ベースの完備性評価をすることが必要となる。

知識ベースの完備性を評価する方法は２つある。動的テストに基づく評価と、静的テストに基づく評価である。前者は、最終的にはユーザがシステムを使用して、満足な結果を得るかどうかを運用テスト時に見つけるものである。満足した結果が得られていないとすると、後者により、その原因は知識の欠如なのか、知識ベースに格納されていても活用されていないのか（知識の未到着、知識の袋小路現象）などを突き止めねばならない。

（４）点検項目(C)・測定項目(M)

(C01)［知識の分解レベル］［モデルの範囲のチェック］知識の分解レベルの詳細度がアンバランスになっていないこと。
例えば、車の故障診断の場合、電気系が冷却系よりも詳細に知識分析書に記述されている場合、本当にそれで妥当なのか。

(C02)［属性の分解レベル］ある属性だけの分解レベルが細分化されて、アンバランスになっていないこと。
細分化が不備である属性は再検討する必要がある。

(C03)［テスト知識］テスト知識の欠落をチェックすること。
テスト知識とは、Ａ or Ｂのとき、ＡとＢのいずれかであるかを判別する知識で、これは専門家にインタビューしないと出てこない知識である。

(C04)［システムの取り扱う対象とする範囲の明確化］取り扱う対象の知識を充分に完備することは困難なので、むしろ何処まで完備したのか、今後どう段階的に発展するのか等を充分把握すること。

(C05)［レベルのアンバランス］ルールの導出レベル、フレームの階層構造のレベルがアンバランスになっていないか、構造ツリー等を表示してチェックすること。

(C06)［感度分析］前向き推論の場合、入力データを変化させること（感度分析）により、知識ベースの完備性を把握すること。

(C07)［完備性のテスト］ドキュメント及びテストにより、完備性を阻害する下記の誤りを検出すること。

- 未参照の属性値はないか。
  属性値に対し、属性値を参照する条件部を持つルールがない。
- 誤った属性値はないか。
  例） lateという値はない。
  LEGAL_VALUE(PROJECT_STATUS,<on-time,behind,ahead>)
  PROJECT(X),PROJECT_STATUS(X,late)
- 欠如ルールはないか。
  オブジェクトの属性がいかなるルールによっても参照されているルールがないことを発見する。
- 未到着な結論部を持つルールはないか。
  ルールの前提部が他のルールに一致する結論部がないか、またはオブジェクトがないことを発見する。
- 袋小路の条件部を持つルールはないか。
  ルールの結論部に一致するルールの前提部がないか、あるいは結論部を満足させるためのオブジェクトがないことを発見する。

(C08)［ルールと事例数の検討1］一つのルールがカバーする事例数が少ない、つまりルールの生成に使用した事例にルールが依存し過ぎているときは、「ルールが一般的でない」と判断すること。

(C09)［ルールと事例数の検討2］カバーされなかった事例数及び余分なルール数が多いときは、ルールの完備性が低いので注意すること。

(C10)［ルールと事例数の検討3］ルールと事例の対応表によって、生成されたルールがカバーしている事例を確認すること。
また、ルールの重要度、事例が誤っていたのではないかもチェックする。

(C11)［未発火ルール］用意した全てのテストデータに対して、発火したルールと未発火ルールを区分して把握すること。
実施したテストの集団で、未発火ルールの割合を知ることにより、どの程度テストが実施されているのかを把握する。

(C12)［未参照の属性値］用意した全てのテストデータに対して、参照・更新した属性値と未参照の属性値を区分して把握すること。
実施したテストの集団で、未参照の属性値の割合を知ることにより、との程度テストが実施されているのかを把握する。

3.1.1.3 知識ベースの一貫性評価

（１）定義

知識ベースの一貫性を阻害する要因が、内在しているか否かを評価することである。

（２）目的と必要性

知識ベースが大規模になればなるほど、知識の一貫性を保つことが容易でなくなる。しかしながら、知識の一貫性の欠けたシステムは類似の要求に対して全く異なる結論を導く可能性もあり、ついには実用性がなくなる。

特に、運用時の知識の更新に対する一貫性評価はより重要である。

（３）説明

知識ベースの一貫性とは、競合した知識がないこと、冗長な知識がないこと、未参照の知識がないこと、知識間で巡回していないことをいう。

また知識ベース内の知識に相互に関連する知識がある場合には、その一貫性を保つことが必要である。特に保守の際に、その関連を無視して、一方のみ更新するというようなことがおこりやすい。

知識ベースのルールをシステマチックに解析し、一貫性の欠陥を見つけ出すことが出来る。例えば、適切でない応答を起こしている原因を見つけるため、トレースしてルールの発火やその順序を追い、知識が如何に不正確で適切でないかを突き止めることが出来る。

なお、知識ベースの一貫性を保つための方法には、

- 知識ベースをルール群に機能分けする
- ルール番号による識別化
- 黒板モデルやフレーム表現を用いて、知識をモジュール化して、知識ベースの構造自体を見通しをよくすること
- ＴＭＳ／ＡＴＭＳ機構を用いて知識ベースの一貫性を保つこと

等がある。

（４）点検項目(C)・測定項目(M)

(C01)［外部環境の変化への対応］対象モデルが、外部の環境に耐えられること。

(C02)［対象モデルの一貫性］［複数専門家の知識の整合性］同一対象に対して、複数の専門家の知識を採用した場合、それらの知識に関して整合性が取れていること。

(C03)［知識の変更］対象とする問題の範囲を変更した場合、それに関係する知識を適切に変更すること。

(C04)［知識の階層性］［知識の継承性］［知識の断片性］応用分野の特性が整理され、その知識の階層性、継承性、断片性などの特徴を整理すること。

(C05)［モジュール化］ルール群を適切にモジュール化し、ルール名またはルール番号の識別に工夫してあること。

これは、ルール数の規模が増えると、全体の構造が見えにくくなりがちであるため。

(C06)［ルールの重複］似たルールが複数のルール集合に分散して存在しないこと。

(C07)［ルールの一貫性のテスト］ドキュメント及びテストにより、ルールの一貫性を阻害する下記の誤りを検出すること。

- 同じルール名はないか。
- 構文的に誤ったルールはないか。
- 冗長ルールはないか。

例）　IF　A & B　THEN　C
　　IF　B & A　THEN　C

- 競合ルールはないか。

例）　IF　A　THEN　B
　　IF　A　THEN　NOT　B

- 包摂ルールはないか。

例）Bは関係ない。
　　IF　A & B　THEN　C
　　IF　A　　THEN　C

- 循環ルールはないか。

①直接サイクル（クローズ内）
例）　ANCESTOR(A,B) and PARENT(B,C) -> ANCESTOR(C,A)

②間接サイクル（クローズ間）
例）　IF　A　THEN　B
　　IF　B　THEN　A

- 不必要な条件項を持つルールはないか。
- 型不一致のルールはないか。

(C08)［ルールの条件部の適切さ］ルールの条件が厳しすぎないこと（特殊化）、逆にルールの条件が緩やかすぎないこと（一般化）。

(C09)［操作容易性］ルールの適用条件が同じような時、条件部(p,q)の兆候は、他のルールの条件部(r,s)に比べてきわめて希な場合は、変更する必要がないのかを検討すること。

例）　IF p,q THEN h1
　　IF r,s THEN h2

(C10)［仮説からの展開度］ある仮説を持つルール群が、他の仮説と比較して極端に少ない場合、本当にそれが妥当であるのかを検討すること。

(C11)［フレームの一貫性のテスト］トキュメント及びテストにより、フレームの一貫性を阻害する下記の誤りを検出すること。

- 非合法のスロットの値はないか。
- 同じ名前のフレーム名はないか。
- スロットの継承の競合はないか。
- 継承パスの循環はないか。

(C12)［整合性の確認］知識ベースを追加、修正、削除を行うときにグローバルな整合性を確認する手段を用いること。

- 追加知識及び修正知識と参照知識との整合性。
- 削除知識を参照する知識はないか。

《評価方法に関する評価》

3114 経験的知識を正当化する方法とその確認

（1）定義

経験的な知識を正当化する手段が良かったかどうかを確認することである。

（2）目的と必要性

経験的な知識とは専門家の“うまいやり方”を用いることによって、問題解決能力を高めることが可能である知識のことをいう。但し、専門家自身を客観的に評価する基準が明確ではなく、経験的知識の評価は困難であるといえる。

しかしながら原理原則や教科書的な知識の少ない対象では、経験的な知識に頼らざるを得ない。それらの知識を何等かの形で正当化する方法を見いだし、その確認をする必要がある。

（3）説明

経験的な知識の評価の基本は、客観的に正当であることを確認することである。

その方法としては、

- 統計的手法による検証
- 複数の専門家による合意形成
- 深い知識から経験的知識が導き出されるのかの検討

等がある。

また、経験的知識を生かすためにも、経験以外の教科書的な知識を可能な限り整備する努力が必要となり、その効果が大きいことが分かってきた。

（4）点検項目(C)・測定項目(M)

(C01)［知識の偏り］知識の獲得に協力した専門家自身の知識の偏りをどう評価するのかは問題であるが、第3者専門家等の意見を入れて充分検討をすること。

(C02)［正当化理由］経験的知識は、それ自身客観的に正当化を図ることが困難であるため、客観的な深い知識に対する資料の整備、その深い知識との比較、一般的に成り立つかどうかの検討などその知識に対して何等かの正当化理由が見あたらないかを検討すること。

(C03)［統計値を利用した正当化］数値データまたは数量化が可能なデータで再現性のある経験的な知識の場合、統計的手法を用いてどの程度の信憑性があるのかを確かめることが出来るので、その検討を行うこと。

(C04)［知識の深層］［経験的知識の適用範囲］言葉上似通っている知識でも、その意味する内容が全く異なることがあるので、この場合どう異なるのかを深く追求すること。

例）故障診断で「急激な温度上昇」と「瞬間的な温度上昇」

(C05)［過去のデータからの検討］過去の実績データがある場合は、そのデータにより経験的知識が裏付けられるのかを検討すること。

(C06)［経験的知識の適用範囲］「経験的知識」＝「ローカルな知識」であるから、その適用範囲について充分注意すること。

《開発／保守方法・技法・ツールに関する評価》

3.1.1.5 知識表現の選択と適切さの確認

（１）定義

知識表現の選択が対象とする問題に適切であったかどうかを確かめることである。

（２）目的と必要性

知識表現は、記述のしやすさ、表現された知識の読みやすさ、保守の手間、実行効率、メモリ効率などに影響を及ぼすので、その選択が適切であるのか否かの評価は必要不可欠である。

（３）説明

エキスパートシステムを開発する際、対象とする問題にふさわしい知識表現を採用しているのか、さらにその表現をどう生かすようにしてあるのかを確認することは重要なことである。

ここでは、ルール表現、フレーム表現、黒板モデルに基づく表現の特徴について述べる。

1．ルール表現

ルール表現には、表3.1.1.5-1に示すような長所、短所、そして留意点がある。

表3 1 1.5-1 ルールの長所、短所

長所	• 個々のルールは作りやすくてわかりやすい。 • 追加や変更が容易である。 • 推論メカニズムが簡単である。
短所	• ルールの相互関係が不明確である。 • 知識の全体像がつかみにくい。 • ルール数が大きくなると処理効率が非常に悪い。 • 人間の知識構造とは異なる。 • 推論に柔軟性がない。
留意点	• 大規模化するとその全体構造が不明確となりがちであり、そのためルールに識別番号を導入したり、モジュール化を図る等の手だてをしなければならない。 • 各種のルールコンパイルによる推論速度の向上。 • 黒板モデル、協調推論を用いたルール群のモジュール化。 • 推論エンジンのトレーサの提供。 • ルールツリーの自動生成。

2．フレーム表現

フレーム表現は、複雑な概念構造の記述や、局面に応じて異なった推論方法を応用したい問題には特に強力である。その反面、知識ベース全体としての完備性や一貫性を管理することが困難になる。特に、付加手続きの乱用はこの点での危

険が大きい。なお、付加手続きの機構（メッセージ交信の機構も）を外してフレーム表現をプロダクションシステムのデータベースとして利用するケースも多い。

また、フレーム表現では、ある典型的な状況や事象を手続きとして表現しておき、これを推論過程で利用することにより、期待駆動型推論(Expectation Driven Reasoning)をモデル化できる。これにより推論方法が固定されず柔軟性のある方式を採ることが出来るが、開発者側の負担が重くなりがちである。このため、知識エディタやトレースなどの開発環境の整備が不可欠である。

フレームの表現は、問題のタイプにより、表3.1.1 5-2に示す3種類の利用形態が考えられる。

表3 1 1.5-2 フレーム表現の利用形態

利用形態	説　　明
フレームシステムを単に事実データベースとして用いる方法	これは、フレーム型データ構造をデータベースとするプロダクションシステムてある。
付加手続きの役割を利用した方法	これは、インヘリタンスとデーモンの機能を備えたフレーム型データベースを持ったプロタクションシステムである。
メッセージ交信による方法	これは、フレームがお互いにメッセージの交信を行い、局面に応じて知識の種類や推論方法を動的に変えながら、全体として最終的な目標へ向かって行われる仕事に適した方法てある

3. 黒板モデル表現

黒板モデルによる表現には、表3 1 1 5-3に示すような長所、短所がある。

表3 1 1.5-3 黒板モデル表現の長所、短所

長所	• 分散処理と作業記憶の分割による推論の高速化。 • 知識源のモジュール化による拡張性、信頼性の向上とメンテナンスの容易性。 • 曖昧さを持つ情報を多様な知識を協調的に用いて解釈可能とする。 • 並列処理の可能性。
短所	• イベント駆動型推論方式（イヘントによる知識源の駆動）は誤った情報の混入や必要な情報の欠落に対して弱い。 • アジェンダ機能による知識源起動では、優先順位ての重要度が低い問題はネグレクトされる可能性がある。

黒板モデル表現は、問題のタイプにより、表3.1 1 5-4に示すような3種類の利用形態が考えられる。

表3.1 1 5-4　黒板モデル表現の利用形態

利用形態	説　　明
イベント駆動型推論方式	この方式では、知識源はブラックボート事象と呼ばれるブラックボード上の特定の変化に反応して活性化される。ブラックボートモニタは活性化した知識源の詳細な条件をチェックし、もし実行可能であれば、その知識源のインスタンスを作り、これをスケジューリング行列に入れる。スケジューラは現在の制御の集点のもとで最適なインスタンスを選択し、これを実行する。
目標駆動型ブラックボードモデル	これは、イベント駆動型推論方式の問題点を克服するため、ブラックボード事象から目標を生成するメカニズムを導入することによって、イベント駆動型推論方式と目標駆動型制御を統合化する方法である。
アジェンダによる知識源駆動	アジェンダとは、未解決のゴールなとを書き並べておく備忘録のことである。この方式てはアジェンダの中から、重要度順（優先順位）あるいは解けそうな順に問題を拾い出していく。

（4）点検項目(C)・測定項目(M)

(C01)［知識表現手法と対象問題の特徴の関連］各知識表現（ルール表現、フレーム表現、黒板モデルなど）の得失を整理すること。

(C02)［知識表現の選択条件］知識表現を選択する基本条件は、専門家の知識が正確に表現できる、記述レベルが統一できる、読みやすい知識表現、保守の手間、実行効率、メモリ効率などであるから、これらをまず充分に検討すること。

(C03)［知識獲得の容易さ］採用した知識表現は、知識獲得がスムースに行える枠組みであること。
日本語の表現が可能、一旦入力したスロットや、ルール内の条件等が再利用出来ること。

(C04)［専門家の認識モデルに基づく知識表現］［応用分野の特性を考慮した妥当な知識表現の採用］［知識表現の理解容易性］知識ベースをレヒューする際、直接ドメインの専門家に見てもらう場合は、専門家にとり理解しやすい知識表現形式であること。

(C05)［知識表現の効率性］［推論機構や外部モジュールによる処理の依存度］
知識表現と推論機構、データベース、外部モジュールとのトレートオフ、親和性を検討して、知識表現の形式を定めること。

(C06)［緻密さのレベル］知識表現の緻密さのレベルは、期待する効果や対象とする問題解決の制約条件を充分満たすこと。

(C07)［知識表現の保守容易性］保守がしやすい知識表現であり、追加、修正が容易に行えること。
知識の一覧、ルールとオブジェクトの関連等が充実しており、知識の更新作業が容易であること。

(C08)［ツールへの依存］ツールに引きずられた知識表現になっていないこと。
ツールを選定する場合に、(C01),(C02)の点検項目について充分検討すること。

(C09)［ルール表現］ルールで表現できるかどうかを充分検討すること。
ルール形式で自然に表現できるものは、以下の通りである。
IF　条件　THEN　結論または行動
IF　原因　THEN　結果
IF　刺激　THEN　反応
IF　相談者の状況　THEN　アドバイス

(C10)［フレーム表現］フレームで表現できるのかどうかを充分検討すること。
自然に表現できるものは、「～の～は、～である」。
また階層構造に合うものは、「～は～の一種である」。

3.1 1.6 知識獲得方法の選択と適切さの確認

（１）定義

知識獲得の方法の選択が適切であったかどうかを確かめることである。

（２）目的と必要性

知識獲得方法の選択を誤ると、充分な知識が引き出せなかったり、収集できなかったりする。また、知識獲得の効率を向上させるため、知識獲得方法や支援ツールの選択が適切であるのかを確認する必要がある。

（３）説明

エキスパートシステムの能力は知識ベースの中に格納されている知識の量と質で決まる。従来から知識獲得方法として専門家へのインタビューがあるが、その困難性が指摘されており、最近ではいろいろな知識獲得支援ツールが開発されている。現在商品化されているツールは少ないが今後出てくるので、ツールの自動化も含めて、以下に知識獲得支援ツールの大まかな選定基準を示す。

①知識獲得支援ツールの選定基準

知識獲得支援ツールの選定は、知識源を何に求めているのか、表３１１６-1に示す区分によって選定する必要がある。

表３１１６-1　知識獲得支援ツールの分類

ツールの種類	区　　分	ツール例
専門知識の獲得ツール	• タスク限定による知識獲得の容易化（Task Specific Tool TST）	CSRL,DSPL,Copernics
	• 知識獲得支援機構の実現（Knowledge Acquisition Tool KAT）	TEIRESIAS,MORE,NEXTRA,MOLE,SALT,SEAR
教科書知識の洗練化ツール	• 知識洗練化機構の実現（Knowledge Refinement Tool KRT）	SEEK,SEEK2
データ群からの知識生成ツール	• Rule Induction Tool RIT	EXTRAN,RULE-MASTER
	• Neural Network Tool NNT	Neural Works
ドメインモデルから知識生成ツール	• Domain Specific Tool DST	G2,IN-ATE(ICAT),Test Bench

②専門家へのインタビュー

専門家へのインタビューにより知識を得る場合、専門家の性格により自分の仕事を実際以上に複雑に述べたり、逆に実際より簡単であるかのように言う場合がある。また、曖昧な質問をすると意図に沿わない知識が獲得される。その結果作成された知識ベースは、専門家の持つ知識構造とも異なるのみならず、知識獲得自体失敗に終わることとなる。

専門家へのインタビューの際の基本的な留意点は以下の通りである。

- 対象とする専門領域の知識を可能な限り事前に修得しておくこと。
- 専門家のおかれた環境や立場を勘案してインタビューすること。
- 自分の理解したことを再確認する目的の質問も行うなど、専門家と自分の間のギャップを埋める工夫をすること。
- わかりにくい点では、質問の仕方を変えるなど角度を変えて専門家の意見を引き出すこと。

（４）点検項目(C)・測定項目(M)

《知識源の妥当性》

(C01)［専門家の選定方法の妥当性評価］知識を提供する専門家の選定が適切であること。

- 開発協力の得られる知識提供者の専門家が身近にいるか。
- 誰もがその人を専門家と認めているか（一般的意見）。
- その専門家の上司または前任者もその人を認めているか。

(C02)［参考資料］参考にした資料が適切であること。

- 古くないか（バージョンアップされているか）。
- 偏りがないか。

(C03)［事前調査の網羅性］［知識の収集］対象とする専門領域の知識を可能な限り収集して事前に修得しておくこと。
深い知識（原理・原則的知識）、浅い知識（経験的知識）、専門家からの知識、文献・資料に基づく知識など。

《知識獲得過程の妥当性》

(C04)［インタビューの心得］専門家のおかれた環境や立場を勘案してインタビューすること。

(C05)［インタビュー戦略］インタビューの戦略が適切であること。

- 事前準備を行ったか（たたき台をもとに行ったか）。
- たたき台に引きずられ過ぎていないか。
- 視覚的に分かりやすい方法を用いて専門家に確認を取ったか。
- 専門家とのコミュニケーションは充分であったか。
- 本来専門家はXを思っていても、Xに近いX’の程度しか言わないので注意すること。

(C06)［知識の分類］診断の知識の場合、仮説（診断結果）、兆候（仮説の発生により生しる出来事や状態）、条件（直接的には仮説の前兆とならないが他の兆候の診断の際に影響を及ぼす状況内の出来事や状態）、リンク（実態間の因果関係の表現に使用）、パス（仮説と兆候を結合する）などに分類して整理すること。

(C07)［関係知識］項目と項目の関係知識を整理したが、まだ曖昧な関係である、専門家が納得できないときは、専門家自信意識になかった問題解決における重要な矛盾があるため、専門家はこれらに気づいて検討すること。

(C08)［知識獲得支援方法］［知識獲得ツール］問題のタイプに適切な知識獲得支援方法またはツールを使用すること。

- Personal Construct Theoryに基づく格子(grid)方法論を用いた自動インタビュー
- ルールの不備な部分の検索を行ない、より正確なルールを生成するため専門家に問い合わせる。

(C09)［知識の精粗］知識の精粗のレベルを統一すること、不足している知識は何かが明確にわかること。また獲得が困難であるところについて、今後の知識獲得手段につき充分検討すること。

(C10)［ＫＥとの協力］ＫＥとの協力を必要とする場合、専門家とＫＥとの役割が適切であること。

(C11)［新しい状況］新しい状況の場合、それに対応できる専門家はいないので、この場合の知識獲得をどうすべきかも踏まえて検討すること。

(C12)［テストケースの収集］テストケースの収集も知識獲得と合わせて、プロジェクトの早期の段階から始めておくこと。
専門家が知識ベースを信頼できなければ、実用化にはならないため、専門家が納得できるように、テストに大いに参加してもらうことが重要である。

3 1.2 問題解決・推論方式の妥当性評価

解説の該当節　　印
3.1.1　知識ベースの妥当性評価
3.1.2　問題解決・推論方式の妥当性評価 3.1.2.1 推論処理の妥当性評価 3.1.2.2 推論時間の妥当性評価 3.1.2.3 推論手法の選択と適切さの確認 3.1.2.4 不確実情報に対する処理技法の適切さの確認 3.1.2.5 学習機能の適切さの確認
3.1.3　結果の妥当性評価
3.1 4　要求仕様に対する達成度評価
3 1.5　知識ベースの保守性評価
3 1.6　ヒューマンインタフェースの妥当性評価
3.1.7　システムの総合評価

（１）定義

問題解決・推論方式がその対象の問題領域において妥当であるのかを評価することである。

（２）目的と必要性

エキスパートシステムは知識ベースと推論機構から構成される。従って、たとえ知識ベースが妥当であっても問題解決・推論方式が妥当でなければ、やはり正当な結果を得ることはできない。特に最近は高次推論など複雑な推論処理が要求されるようになり、その影響は増大しつつある。

《プロダクトに関する評価》

3121 推論処理の妥当性評価

（1）定義

推論処理がその対象の問題領域において妥当であるのかを評価することである。

（2）目的と必要性

問題解決推論方式が妥当でないと、たとえ知識ベースが正しくても解ける問題が解けなくなってしまう、または正しい答えが求められない等の可能性が高く、その評価は重要である。

（3）説明

エキスパートンステムの初期の時代には、推論処理は単純にしておき知識の側で高い機能を実現し得ると考えられていた。しかし、最近では類型的タスクに代表されるように問題解決を重視する必要性が認識されている。

このため、推論処理も次第に複雑なものが要求されており、その論理的な正しさを確認することが重要となっている。

推論処理については、一般のプログラムとの共通点も含め、以下の留意点を考慮して評価する。

①知識表現との整合性（分担の明確化）

②動作の透明性

③停止性

④モジュール性

⑤動作モードと状態遷移

⑥ＡＩ言語上の各問題点

- メモリ管理
- 再帰呼び出し
- 動的なプログラムの書換え
- インタプリタ／コンパイラ

（4）点検項目(C)・測定項目(M)

(C01)［試行錯誤的処理］構築するンステムは、本当に試行錯誤的に処理を行う必要があるのかを確認すること。

(C02)［モジュール化］［メタ知識］専門知識、推論戦略知識、推論エンンンの分離が明確に行われていること。

- 推論戦略知識（メタ知識）は、専門知識または推論エンンンに紛れ込んでしまい、見通しが悪くなっている場合が多い。

(C03)［推論に要求される厳密度］［曖昧性の取扱い］論理的な推論、確率的な推論、曖昧な推論の仕方により、推論過程が妥当であるのかは、その厳密度が異なるので注意すること。

(C04)［質問のタイミング］システムから出される質問のタイミングから推論処理の妥当性を確認すること。

(C05)［推論動作の透明さ］推論動作と状態が透明で分かりやすいこと。

- 特に、知識の競合解消方式、不確実性の処理方式、非単調な推論の場合。

(C06)［推論過程のトレース］推論過程をトレースし、推論で利用される知識及び中間結果が妥当であること。

(C07)［標準機能の利用］推論処理の動作、状態を分かりやすくするため、標準的な推論機能またはツールを用いて構築すること。

(C08)［推論の停止条件］解が存在しない場合、フェイルした場合、次のタスクに移る条件など、推論の停止条件を確認しておくこと。

(C09)［説明機能の充実］システムが誤った場合でも、利用者がなぜ誤ったのかの判断が可能となるように、推論過程を説明する機能の充実を図ること。

3.1.2.2 推論時間の妥当性評価

（１）定義

問題解決に要した推論時間が妥当であるのかどうかを評価することてある。ここでは、トータルなレスポンス時間の評価も含める。

（２）目的と必要性

エキスパートシステムの処理時間には推論処理の効率の影響が大きい。従って、推論時間の妥当性はエキスパートシステムの実用上の重要な要素である。

（３）説明

エキスパートシステムでは、推論の結果を得るまでに要する推論時間は、知識ベースの内容と推論機構の性能に依存する。推論に必要な時間を具体的に述へると、プロダクションシステムであれば認識－実行サイクルの時間、フレームシステムであれば発生や書換えに要する時間や継承関係をたぐる時間などであり、これらが評価対象となる。

以下の留意点を考慮して評価する。

①対話性
②知識ベースの規模の考慮
③リアルタイム性の要否
④ガーベジコレクションの有無、頻度

（４）点検項目(C)・測定項目(M)

(C01)［専門家のイメージ］推論時間は、専門家がイメージしている処理時間に合っていること。

(C02)［推論時間］特に、推論時間が長いと感じる場合、
- 多段の推論にならないようにすること、
- システムとの対話性がよいこと、
- フレームの継承関係の参照時間、
- ルール群当りのルール数が多くないこと、
- ガーベージコレクションの頻度、
- 同じ計算の繰り返しなどの冗長な記述がないこと、
- 手続き処理をルールで記述していること、
- フレームの乱用等

を充分検討すること。

(C03)［監視機能］推論時間（または処理時間）が厳しいときは、タイマーによる監視機能を備えていること。

(M01)［枝刈り知識］前向き推論ではコンフリクトセットのトレースをとり、コンフリクトの多い状態を分析して枝刈り知識が得られないかを検討すること。

(M02)［推論エンジンの処理時間］推論時間を測定した結果、推論エンジンに疑問がある場合は、前向き推論、後向き推論などのプリミティブなレベルで処理速度を測定する。

(M03)［推論時間の増加オーダー］問題の規模増加に対する推論時間の増加オーダーを解析的に調べる、または測定する。

(M04)［生成検査法の場合］生成検査法の場合は、生成される探索ノート数、単位ノートの処理時間を測定する。

(M05)［ﾌﾟﾛﾀﾞｸｼｮﾝｼｽﾃﾑの場合］プロダクションシステムの場合は、認識ー実行サイクルの時間を測定する。

(M06)［ﾌﾚｰﾑｼｽﾃﾑの場合］フレームシステムの場合は、発生や書換えに要する時間や継承関係をたぐる時間を測定する。

《開発／保守方法・技法・ツールに関する評価》

3 1.2 3 推論手法の選択と適切さの確認

（１）定義

推論手法の選択が適切であったのかどうかを確かめることである。

（２）目的と必要性

推論手法の選択を誤ると、知識ベースをうまく扱えない、知識ベースを構築する際に手間がかかる等の問題が生ずるのみならず、当然期待する結果が得られぬ可能性が高くなる。

（３）説明

エキスパートシステムの基本機能は、知識表現と推論手法とで決まる。特に、設計やスケジューリングの合成型問題では、推論の性能において推論手法の占める比重が高いことが指摘され、開発の初期の段階において推論手法の選択が重要である。

開発の初期段階において、対象問題に応じて以下の項目を検討し、推論手法を定める必要がある。

①問題解決パラダイム（生成検査法、階層的生成検査法、抽象化、分解･統合、制約伝播、トップダウン精密化、拘束最小化、依存関係に基づく後戻り）

②演繹的推論手法　（プロダクションシステム、述語論理）

③高次推論手法　（不確実性推論、デフォルト推論、仮説推論、分散協調型推論、定性推論、事例ベース推論）

以下の留意点を考慮して評価することになる。

①推論手法の複雑さと知識獲得の負荷とのトレードオフ

②システムの透明性

③デリバリ環境

（４）点検項目(C)・測定項目(M)

(C01)［問題解決のモデルやパラダイムの適切性］問題解決のモデルやパラダイムは、対象とする問題に対して適切であること。

- 部分問題（基本タスク）への分解
- 競合解消の戦略
- バックトラック処理と解のグローバルな最適性
- 動作の適切性（動作モードと状態遷移）

(C02)［基本的な推論手法］対象となる問題に対して妥当である、基本的な推論手法を検討すること。

- 基本的な推論手法については、上記の説明を参照のこと。

(C03)［知識獲得の負荷］推論手法の複雑さと知識獲得の負荷とのトレードオフを考慮すること。

(C04)［冗長な構造］定量値がとれるのに定性推論で行っているなど、冗長な構造になっていないこと。

(C05)［バックトラック機構］効率のよいバックトラック機構になっていること。

(C06)［推論手法選択の考慮点］機能性、信頼性、効率性、使用性、段階的発展性を考慮して推論手法を決めること。

また、ツールを選定するときには、そのツールの持つ推論手法を充分検討すること。

3 1 2 4 不確実情報に対する処理技法の適切さの確認

（１）定義

不確実な情報に対する処理技法の適切さを確認することである。

（２）目的と必要性

エキスパートンステムに対する期待の一つに、不確実な情報に基づく推論があげられる。従来のベイズ確率のほか、MYCINの方法やDempster-Shaferの方法、及びファジィ推論に基づく方法などが使われるようになってきている。しかし、安易に不確実性を導入すると、推論結果の妥当性の保証が困難、知識ベースのチューニングに要する負荷大、保守性の低下、等といった問題を引き起こす場合がある。従って、対象とする問題に応して不確実情報に対する処理技法が適切であるのかを充分確認する必要がある。

（３）説明

以下の留意点を考慮して評価する。

①対象問題に内在する不確実性の明確化
②不確実性の性格付け（基数的、序数的、メンバーンップ的など）
③不確実性の局在化
④期待される出力と処理技法との整合性
⑤不確実性を用いないアプローチの検討

（４）点検項目(C)・測定項目(M)

(C01)［不確実性導入に伴う問題点］［結果の妥当性の保証低下］不確実な情報を扱うと、推論結果の妥当性の保証が困難である、知識ベースのチューニングに要する負荷が大きくなる、保守性が低下する、といった問題かあるので注意すること。

(C02)［基本的な技術］基本的な技術を検討すること。

- ベイズ確率
- Dempster-Shafer理論
- ファジィ理論
- 確信度

(C03)［不確実性の局所化］不確実性を用いないアプローチまたは用いる場合でも不確実性の局所化を検討すること。

(C04)［不確実性の意味］対象問題に内在する不確実性の意味を明確化すること。

(C05)［確信度の性質］確信度の表現及びその操作が、当該問題領域で扱っている確信度の性質や慣行と合っていること。

(C06)［整合性］期待される出力と処理技法との整合性を充分検討すること。

(C07)［根拠付け］不確実な情報の根拠を確率論等によって意味付けること（たたし、この作業は難しい）。

3 1.2.5 学習機能の適切さの確認

（１）定義

学習機能の活用が適切であったか否かを確認することである。

（２）目的と必要性

学習機能とは、ここでは問題解決の過程で問題解決能力の向上を図る機能あるいは知識獲得の自動化を促進する機能であるとする。学習機能の利用を誤ると、思ったほど問題を解く能力が向上しない、知識の抽出能力が十分でないなど、期待した効果が得られぬとともに効率上の影響も出てくる可能性があり、評価する必要がある。

（３）説明

学習機能は、①モデルからの演繹的学習と②データからの帰納的学習に大別される。

①のアプローチは、より基礎的な知識から、与えられた問題を効率的に解決できる知識を再構成する学習方式であるが、解けなかった問題が解けるようになる（知識レベルの）学習ではなく、計算効率が向上する（記号レベルの）学習である。今までに、説明に基づく学習（ＥＢＬ）や知識コンパイルなどの技術が提案されている。本アプローチの利用方法としては、例えば、設計問題において、設計理論は予め整備されている状況で、具体的な設計問題を解決したプロセスから、設計問題を効率的に解く知識を抽出し蓄積していくことなどが考えられる。

一方、②のアプローチは、与えられたデータ群から規則性を発見して、概念を獲得する学習方式であり、知識レベルの学習として位置づけられ、バージョン空間、ルールインダクション、ニューラルネット、事例に基づく推論（ＣＢＲ）などがこのアプローチに入る。また、最初にある知識群を与え、②の技術を利用して、逐次、知識を洗練化するアプローチも存在する。本アプローチの利用方法としては、例えば、故障診断問題において、経験的な故障診断ルールは未整理だけれども、故障データが豊富に蓄えられている場合には、そのデータから故障診断ルールを帰納的に学習することなどが考えられる。

しかしながら、これらの学習技術は、既に成熟した技術ではなく、発展途上にある技術であるため、エキスパートシステムへの利用の期待は高いものの、その利用方法には十分注意する必要がある。

具体的な評価としては、対象システムと以下のような現状技術で可能な学習機能との関連性及び実現の度合をチェックする。

①モデルからの演繹的学習
- 説明に基づく学習（ＥＢＬ）
- 知識コンパイル

②データからの帰納的学習
- バージョン空間
- ルールインダクション
- ニューラルネット
- 事例に基づく推論（ＣＢＲ）
- 知識洗練化

（４）点検項目(C)・測定項目(M)

(C01)［適用条件］［実現可能な基本技術の検討］学習技術の適用条件を十分検討すること。

以下に、各学習技術の適用条件の１つの指針を示す。

ＥＢＬと知識コンパイルにおいては、獲得される知識の記述の詳細さから検討して、その領域の基礎理論が十分な詳細さを保持することが必要である。例えば、定量的な記述を含む知識を獲得することが目標であるのに、領域の基礎理論が定性的にしか記述できなければ、本学習技術は適用すべきではない。

バージョン空間とルールインダクションは、基本的には、データが豊富に存在する線形分離可能な問題に適用すべきである。但し、データが豊富であることは、単に、データが多く存在するだけでなく、データを記述する特徴量が多く存在する（例えば、多くのセンサー値により故障データが表現されている）ことも意味する。また、問題の線形分離可能性を事前に見極めることは困難であるかもしれないが、雑音が少ないことが１つの目安である。

一方、ニューラルネットは、基本的には、データが豊富に存在する線形分離不可能な問題に適用すべきであり、バージョン空間やルールインタクションより能力は高いと言えるが、学習時間は長くなることを留意しておく必要がある。

また、ＣＢＲは、１つの経験が１つのデータで表現されるよりは、複合データ（ケース）で表現され、そのケースが豊富にある問題に適用すべきである。

さらに、知識洗練化は、少量の経験的知識あるいは不確定度の高い知識と豊富なデータがある問題に適用すべきである。

(C02)［学習機能の利用と問題点］学習機能を用いることの利点及び学習機能を実現する上での問題点等を充分検討すること。

(C03)［既存知識との整合性］既存知識との整合性の保証を確認すること。

(C04)［起動タイミング］学習タスクを起動するタイミングが適切であること。

(C05)［学習機能による処理時間の短縮］学習により効率的な問題解決が図られること（演繹的な学習機能）。

(C06)［知識の増加］学習によって知識が増えると、学習した知識も探索の対象となり、最終的にその部分の探索が役に立たない場合には不要なバックトラックが増加し、効率が低下するので注意すること。このため、学習した知識の使用の制限などを検討すること。

(M01)［学習前／後の性能比較］学習例題、テスト例題を用いて、学習前と学習後の処理能力（推論結果の精度と処理スピート）を測定すること。

3.1.3 結果の妥当性評価

解説の該当節　　印
3 1.1　知識ベースの妥当性評価
3 1.2　問題解決・推論方式の妥当性評価
3.1.3　結果の妥当性評価 3.1.3.1 正解の評価 3.1.3.2 解の網羅性評価 3.1.3.3 解の最適性評価 3.1.3.4 解の評価方法の適切さの確認
3 1.4　要求仕様に対する達成度評価
3.1 5　知識ベースの保守性評価
3 1 6　ヒューマンインタフェースの妥当性評価
3 1 7　システムの総合評価

（１）定義

エキスパートンステムから出力された結果が、その対象の問題領域で正しいか否かを評価することである。

（２）目的と必要性

如何なるシステムにおいても結果の妥当性を確認することは重要てある。特に、エキスパートシステムにおいては完全なテスト手法や十分なテストデータが得られぬケースも多く、結果の評価を厳格に行う必要がある。

（３）説明

エキスパートシステムにおいては、その解が正しいのかを確かめる方法がなかったり、あってもそれを客観的に評価することが出来なかったりする。従って解の妥当性を確かめるには、まず妥当性を確かめる手段があるのか、もしないならば工夫して手段を設定せねばならない。正解がわからない場合は、知識を提供した専門家に正解を提示してもらい、開発担当者、マネジャを含めてその解が妥当であるのかを前もってレビューすることが必要である。なお、この評価項目は、問題解決・推論方式、知識ベース、要求仕様と密接に関わり合う。

《プロダクトに関する評価》

3 1.3 1　正解の評価

（1）定義

テストに使用したテスト件数の内、何件正しい結果が出たのかの割合を評価することである。

（2）目的と必要性

解の正解率はシステム利用上の最大の目安である。ユーザはこれによって当該システムの使用方法、使用形態ないしは実用の可否の判断をすることができる。

（3）説明

システムの正解率が具体的、かつ客観的に測定出来るならば、システムのカバーする範囲、システムの性能が明確となるため、まず測定可能になることが大切である。

また、エキスパートシステムの場合、過去の事例が少ない、専門家の経験的な知識を重視するなど、きわめてテストする環境が悪いのが一般的である。従って、従来の情報処理システムと同様の正解率の概念では適切でないケースも考えられるため、ユーザ、マネージャ、専門家の間で、正解率の定義につき予めコンセンサスを得ておくことも重要である。

最終的に専門家が判断を下すシステムでは、必ずしも全ての場合に対して正解を出さなくてもよい。非常に複雑で例外的なケースは専門家に任せ、一般的なケースのみ自動的に処理することで、専門家の負担はかなり軽減される。また、サービスのスピードアップや、専門家の工数削減によるメリットが、不正な解によるリスクを上回る場合には、完全な正解を与えなくても充分実用的なシステムであるといえる。

一方、システムが提示した解を修正することが非常に困難な場合や、不正な解によるリスクが非常に高い場合には、正解の割合は非常に高いものでなければならない。

評価をする際、100％の結果の正解率を追い続けると、いつまでたっても実用化システムにはならない。60～70％の正解率でも以上のことを考慮して使用に耐え、役立つものがあればよい。最終的には人間の判断に頼るシステムが実務的であるともいえる。

以下の留意点を考慮して評価する。

①ユーザからの要求

- エントユーザが使用するシステムであれば、解の正解率を高めなければならないし、結論に至った過程を充分に説明する機能がなければならない。
- 専門家が使用する場合は、誤った場合でもなぜ誤ったのかが判断つくことが多いので、解の正解率がシビアであるにこしたことはないが、むしろ、推論過程を説明する機能を充実すべきである。

②専門家との比較

- 知識を提供した専門家は、予め正解のパターンを作成しておき、開発担当者とレビューをしておくこと。
- 複数の専門家が結果に対し検討するときは、必ず共通見解を出して、合意形成を得て行うこと。

③限界値テスト

- 通常想定できるデータに対しての結果のみをテストすれば、当然解の正解率は良くなる。従って、希にしか発生しないデータ、例外的なデータも含めて限界値テストをする必要がある。しかし、正解率の計算に際しては、このことを充分に考慮する必要がある。

④正解のパターンを充分用意すること

- 意義のあるテストデータは充分あることが当然望ましい。そして予め用意されている正解があることが前提である。

⑤テストケースが異なると予期しない振舞いをする

- テストデータの変更や追加をすると、今までのデータではうまくいっていても、変更後はうまくいくとは限らない。その場合には、知識ベースや推論過程に問題がある可能性もあり充分な追求が必要である。

（４）点検項目(C)・測定項目(M)

(C01)［正解に関するコンセンサス］［正解の定義］正解とは何かを前もって定義しておくこと。

- 正解の定義について、予めコンセンサスを得ておくことが必要である。
- 正解として例えば「確信度が第３位までの故障原因の中に真の原因が含れていること」というように具体的な定義を与えておくこと。

(C02)［正解の事前準備］知識を提供した専門家は、予め正解のパターンを作成しておき、開発担当者とレビューをしておくこと。
動的なテストデータは、実際の状況と同しに作るのは難しい。

(C03)［限界値テスト］［正解率の意味］通常想定できるデータに対しての結果のみをテストすれば、当然解の正解率は良くなる。従って、稀にしか発生しないデータ、例外的なデータも含めて限界値テストをすること。
しかし、正解率の計算に際しては、このことを充分に考慮する必要がある。

(C04)［オンラインデータ］［テストデータ収集機構］オンラインで多量なデータが入力され、その時系列的変化が判断基準としているようなエキスパートシステムではマニュアルで検証されたテストデータを作成することは事実上不可能であるため、対象プラントからテストデータを収集できる機構を準備しておく必要があること。

(C05)［要求仕様］要求仕様で記述した正解率以上であること。
正解率の基本的な評価基準は、予め要求仕様で記述されており、その内容を満足しているのかどうかで評価する。

(C06)［正解の吟味］解は、本当に正解としてよいのかを吟味すること。

- 専門家の解と同じか。
- モデルを反映した解が得られているか。
- 事例の生じる頻度を考慮して、過去の事例や今後起こり得る事例と比較すること。
- 用意したテストデータを流した結果、解は満足が行くか。
- 用意したテストデータは解の正解性をテストするのに充分か。
- 正解率が低すぎるときは、例えば「ルールが一般的すぎる」、「誤って事例を分類する確率が高い」ことがわかる。
- 同じ原因でも求まる解が異なる場合は、問題戦略の吟味が必要である。
- 本来求めるべき解に到達しているのかを吟味する必要がある。

例：故障原因を追求するものと故障箇所が分かればよいものとに区別される。

(C07)［合意形成］複数の専門家が結果に対し検討するときは、必ず共通見解を出して合意形成を得て行うこと。

(C08)［無意味な結果］専門家が設計する場合に比へて、無意味な結果の候補を生成する場合があるので注意をすること。

(C09)［解の精度の意味］現象が頻繁に起こる問題の場合は、正解が多数発生するので、一見解の精度が上がったかのようになるので注意すること。

(C10)［確信度］確信度が入ると正解かは、よく判別できなくなるので注意すること。

(C11)［大きな探索空間］取り扱う探索空間が大きいと専門家と言えともフォローが出来なくなる場合があるので注意すること。

(C12)［解の精粗の要求］評価する部門により求める解に対する精粗の要求が異なるので注意すること。
例えば、設計部門、品質管理部門は詳しく（例えば、故障原因の追求）、サービス部門は荒っぽくてもよい（修理できればよい）。

(C13)［問題解決の仕方、流儀］最初は1人の専門家によって構築し、レビューする段階で複数の専門家が入る場合、その問題解決の仕方や流儀などが異なるので、正解について充分検討すること。
何が正解なのかは、各立場によって異なるので注意すること。

(C14)［回帰テスト1］知識や推論系の保守にともなって、これまで正解の出ていたテストケースで正解が出ない場合があるので注意すること。

(C15)［データ入手のコスト］解を得るのに必要な検査・点検などデータを入手する手間や費用が、かかり過ぎていないこと。

(C16)［回帰テスト2］テストデータの変更や追加をすると、今までのデータではうまくいっていても、変更後はうまくいくとは限らないので注意すること。

(M01)［正解率の測定］正解率＝正解数／総テスト問題数　（ある時点、トレンド）を測定すること。

- 分析型の場合には、解の評価は正解率を用いる。

(M02)［JIS Z9002による方法］計数規準型一回抜取り検査（不良個数の場合）
JIS Z9002の方法により、合格・不合格の判定を下す方法。
全ての解が正解ではなくてもよいとき、またはある程度テストがすんだ段階で、併行テストまたはフィールトテストを行う際に有効な方法である。
（方法）

- 品質基準を決める（正常、バグあり）
- P0（なるべく合格させたいロットの不良率の上限）、P1（なるへく不合格させたいロットの不良率の下限）の値を指定する。
- テストデータの数と合格判定個数を求める。
- テストする。
- 合格、不合格の判定を下す。

3 1.3.2 解の網羅性評価

（１）定義

正しい解が複数求められるへき問題の場合、どの程度網羅的に解を得ることが出来るのかを評価することである。

（２）目的と必要性

解の網羅性は、結果に対して正しい判断を行うためにも必要不可欠な条件である。また大規模なシステムにおいては単に網羅性を上げるのではなく、効率との兼ね合いで最適な網羅性を判断することも重要となる。

（３）説明

合成型問題（設計問題、計画問題）においては、基本的な問題解決パラダイムが生成検査法であることがあるが、この場合候補となる複数の解を生成し、それを検査によって絞り込むアプローチをとる。従って、生成時には、候補解が漏れなく上がっていることが望ましい。候補の生成問題は、探索問題に帰着する。即ち、探索空間が小さければ網羅的な探索が可能て解の網羅性がよいが、探索空間が大きくなるにつれ、探索の効率向上をさせるため、問題解決戦略を導入したり、可能性の低い解候補の探索を排除するために枝刈り規則を利用することが多い。

従って、計算量、要求仕様、解の網羅性の３者が密接に関わりあっている。

以下の留意点を考慮して評価する。

①網羅性の目標

②仮説の生成とその矛盾

③生成検査

（４）点検項目(C)・測定項目(M)

(C01)［網羅性の目標］計算量や対象とする問題の特性から、取り扱える対象範囲、解の単一性、解の複合性など、どの程度の網羅性を目標とするのかを予め充分検討すること。

計画型ノステムでは通常、組合せ爆発の問題があり、この問題をどう克服するかが本質的な課題となる。

(C02)［解候補の生成］解候補を生成して、その時点でその解候補が妥当てあるのかどうか不明のときは、仮にそれを選択し仮説として扱うこと。

もとの仮説と矛盾が生じなければその範囲内で正当化され、矛盾が生したときは依存関係に基づいて後戻りする。従って解の網羅性を評価するに当たっては、問題解決パラダイムを充分検討すること、及びこのパラダイムで取り扱われる知識を充分検討しなければならない。

(C03)［解候補の検査］解候補の生成に続いて検査が行われ、もし不適切であればその解候補は撤回され、後戻りして別の解候補を探索すること。

従って問題解決パラダイムで取り扱われる知識を充分に検討することが必要である。

(C04)［解の順番性］より良い解が順番に求められてくること。
これを専門家に評価してもらう方法として例えば、次のものがある。

- エキスパートシステムが出力した解の集合を専門家に評価して順序付けをしてもらう。もし、この順序付けと出力結果の順とが一致していればＯＫである。
- 評価値＝Σ（出力の順番－専門家の付けた順番）^{2}
  この値がゼロに近いほど良い（ゼロは完全に一致）

(C05)［専門家の解］専門家の解を本当に導出できること。
点検のやり方として以下の方法がある。

- 専門家の解（模範解答）をインプットして、エキスパートンステム内にある制約知識を使って制約違反チェックを行う（この様な機能をエキスパートシステム内に組み込んでおく）。全て、制約にかからなければ、このエキスパートシステムでも同様の解を導くことが出来るものと判断する。
- もし、制約にかかり、エキスパートンステムをある程度走らせても解が出てこないのならば、評価関数が悪いのか、知識の優先度のチューニング不良と考えて調整する。
- 模範解答の検証が、エキスパートシステム内の制約知識に抵触し停止したら、模範解答が誤っているのか、制約が厳しいのか、制約緩和の優先度のチューニング不良か、制約緩和機構の不良かのいずれかが原因であると考えられるため、焦点を絞って検討する。

(C06)［回帰テスト］同一の問題に対して、知識ベースの変更後、これまで出力されていた解が出力されること。

(C07)［実用上の適切さ］解の網羅性の度合は、実用上適切であること。
網羅度が小さすぎたり、反対に不必要に大きいためその結果推論時間が長すぎたり、ユーザが混乱したりすることはないのかをチェックする。

(M01)［推論時間の増加オーダー］網羅度増加に対する推論時間の増加オーダー（測定、解析的）を測定する。
ただし、推論時間の妥当性評価とのトレードオフを考慮すること。

3.1.3.3 解の最適性評価

（１）定義

最適解を求めることが目的である場合、対象とする問題の領域において、得られた解が最適であるか否かを評価することである。

（２）目的と必要性

解の最適性が保証されないとユーザ自身がその最適性を調へなければならない。また、理論的に完全な最適性を求めることは無理な場合であっても最適度が高いほど、そのシステムの有用性が確認されたことになる。

（３）説明

対象とする問題が閉空間を取り扱い、システムからの結果がシヒアでなければならない問題の場合は、解が最適性を満足しなければならない。解が最適であることを保証するためには、最適解に対する評価基準が重要である。

設計問題、計画問題は、一般に最適解は未知である。従って、専門家が作成した過去の設計案・計画案との比較、またはＯＲ手法（線形計画法、動的計画法、整数計画法など）での解と比較してどちらが、より制約条件、評価基準値を遵守しているのか等を検討することが必要である。

（４）点検項目(C)・測定項目(M)

(C01)［最適性の条件］最適解を必要とする問題の場合は、最適性が保証される問題解決方法を取ること。

(C02)［評価関数がある場合］評価関数がある場合は、エキスパートシステムに組み込まれた評価関数の値により、最適性を検討すること。

(C03)［評価関数がない場合］評価関数がない場合は、専門家の解と比較すること。

- 専門家が作成し、整備した解とエキスパートシステムからの解を比較せよ。
- 現場のユーザは、解の質について満足しているのか。
- 得た解ばかりでなく、解を導き出した過程を専門家に説明し、専門家と合意が得られているか。
- 熟練度の高い専門家の解とのずれ、的中率、誤り指摘率
- 専門家、中級者、初心者とシステムとの比較

(C04)［条件の変化への対処］専門家が理解しにくい評価関数に取り組むと、ある条件のもとでは、よりよい条件が得られても、条件の些細な変化には全く対処できなくなる恐れがあることに注意すること。

(C05)［他の方法との比較］他のシステムまたはＯＲ等の手法と比較が可能な場合は、どちらがより最適性基準を満足しているのかを検討すること。

(C06)［局所解］局所的な解になっていないこと。

(C07)［多目的な評価］エキスパートシステムは、多目的な最適解を持つことが多いので、種々の観点から評価すること。

(C08)［コストとのバランス］解の最適性は推論時間など解を得るのに必要なコストとのバランスを考慮して評価すること。

例え、最適な解が得られたとしても、そのためにコストがかかりすぎる場合は検討しなければならない。

《評価方法に関する評価》

3134　解の評価方法の適切さの確認

（1）定義

結果の妥当性を評価する手段が適切であるか否かを確かめることである。

（2）目的と必要性

評価方法を誤ると当然正しい評価が行われない可能性が高い。適切な評価方法がとられたか否かの確認は、評価の基本要件である。

（3）説明

結果の妥当性を評価する方法は大きく分けると専門家による評価と何らかのツール、手法、データを用いて評価する場合がある。

［人間による解の確認］

①知識提供の専門家による評価

②第3者専門家による評価

③ユーザによる評価

④レビューアによる確認

［データによる解の確認］

①知識の出典、結果の根拠の確認

②過去の事例による対比テスト

［手法による解の確認］

①統計的手法を用いて分析

②プロトコル分析

［確認用ノールによる解の確認］

①数値シミュレーション

②ハードウェアンミレータ

③実装置による疑似故障

（4）点検項目(C)・測定項目(M)

(C01)［評価方法］どの評価方法を用いるのがよいのかを予め検討すること。

①人間による解の確認

- 知識提供の専門家による評価（専門家の能力と比較）
- 第3者専門家による評価（多数の専門家と比較）
- ユーザによる評価
- レビューアによる解の確認

②データによる解の確認

- 知識の出典、結果の根拠の確認
- 過去の事例による対比テスト（正しい過去の結果との比較）

③手法による解の確認

- 統計的手法やOR手法を用いて解析
- 事象はどのくらいの頻度で発生するのか、他の事象との相関はどのくらいなのかを統計的手法を用いて分析。
- システムの収束状態を管理するためのバグ収束管理図。
- 手法を用いたときの評価結果と人間に基づく評価結果とを統計的に比較する。
- プロトコル分析
  人間の心の動きを内観報告する方法。

④確認用ツールによる解の確認

- 数値シミュレーション、ハードウェアシミュレータ、実装置による疑似故障
- 専用ツール（遅延解析ツール等）による評価

⑤フィールドテストによる解の確認

- 初期の段階から、実システムとして位置つける場合
  エラーがあってもコスト損失が少なく危害が少ない場合に有効。
- 併行運用をする場合
  妥当性検査(Validation)を完全にする場合に有効。

(C02)［解の性格］得られるへき解の性格について検討すること。

- 専門家が利用するのか、非専門家が利用するのか。
- 可能解（許容解）なのか最適解（厳密解）なのか。
- 解の正解率は完全でなくてもよいのか。
- 解の網羅性はどの程度であればよいのか。
- パターン解（予め用意が出来る典型的な解）であるか。

(C03)［結果の品質特性］結果や結論に対して検査・評価すへき品質特性を考慮して、解の正解性を検討すること。

- 一貫性(Consistency)
  知識ベースのどのルールが使われようと、どんな推論経路をとろうと、同じ入力データに対し、どの程度同じ結論に到達するかを表す度合。
- 完備性(Completeness)
  エキスパートンステムが対象としている問題領域で正しいと考えられている結論すへてが、どの程度導出されるかを表す度合。
- 健全性(Soundess)
  導出された結論すべてが、どの程度正しいかを表す度合。
- 精度(Precision)
  入力データがほんの少し変化した場合に、出力がどの程度変化するかを表す度合。
- 応答時間性(Response time)
  要求された時間内でどの程度結論を導き出せるかを表す度合。
- 入力順序応答性(Sequence independense)
  データの入力順序を変えても、導き出される結論がどの程度変化しないかを表す度合。
- 強固性(Robustness)
  誤りのあるデータ、不正確なデータ、または一部のデータが欠落していても、導出される結論がどの程度正しい結論であるかを表す度合。

(C04)［評価者］解の評価を誰がどのように行うのか、知識獲得の段階て決めて準備しておくこと。

(C05)［ユーザの同意］問題解決に対する標準的な方法論（解法）が存在しない問題だとすると、ユーザに解の妥当性の同意を取りつけること。

3.1.4 要求仕様に対する達成度評価

解説の該当節　　印
3.1.1 知識ベースの妥当性評価
3.1.2 問題解決・推論方式の妥当性評価
3.1.3 結果の妥当性評価
3.1.4 要求仕様に対する達成度評価 3.1.4.1 要求仕様内容に対する評価 3.1.4.2 要求仕様に対する達成度評価 3.1.4.3 要求仕様設定手段の確認
3.1.5 知識ベースの保守性評価
3.1.6 ヒューマンインタフェースの妥当性評価
3.1.7 システムの総合評価

（１）定義

エキスパートシステムが要求仕様で規定した要件をどの程度達成しているのかを評価することである。

（２）目的と必要性

要求仕様は、ユーザニーズを規定したものである。要求仕様に規定されている要件を達成していないシステムは当然ユーザの満足度が得られず、実用されない可能性が高い。

（３）説明

要求仕様に対する達成度評価は、エキスパートシステムの開発フェーズごとに作成される要求仕様に依存する。図3 1 4-1 にエキスパートシステムの開発フェーズと要求仕様の関係を示す。エキスパートシステムの開発では、各開発フェーズにおいて要求仕様書が作られる。初期段階で不十分であった仕様検討が、プロトタイピングのフェーズが終了した段階で詳細まで決定できるようになる。従って、図3.1.4-1 中の実用版要求仕様書が通常のウォーターフォール型のシステム開発における要求仕様書に相当する。この要求仕様書は実現性が充分保証されたものであり、この段階でンステムの詳細設計が可能となる。

<table>
<tr><td rowspan="2">開発ﾌｪｰｽﾞ</td><td rowspan="2">調査・計画（探査型ﾌﾟﾛﾄﾀｲﾋﾟﾝｸﾞ）</td><td colspan="2">プロトタイピング</td><td colspan="2" rowspan="2">実用ｼｽﾃﾑ開発</td><td rowspan="2">運用テスト</td><td rowspan="2">運用・洗練化</td></tr>
<tr><td>実験型ﾌﾟﾛﾄﾀｲﾋﾟﾝｸﾞ</td><td>発展型ﾌﾟﾛﾄﾀｲﾋﾟﾝｸﾞ</td></tr>
<tr><td rowspan="2">要求仕様</td><td rowspan="2">企画提案書（第0版要求仕様書）</td><td rowspan="2">第Ⅰ版要求仕様書</td><td rowspan="2">第Ⅱ版要求仕様書</td><td colspan="2">改善</td><td rowspan="2">――</td><td rowspan="2">変更要求仕様書</td></tr>
<tr><td colspan="2">再開発</td><td colspan="2">実用版要求仕様書</td></tr>
</table>

図３１４-1　エキスパートンステムの開発フェーズと要求仕様書

《プロダクトに関する評価》

3141 要求仕様内容に対する評価

（1）定義

要求仕様として記述すへき項目が漏れなく記載され、その内容が充分にかつ理解しやすく記述されているか否かを評価することてある。

（2）目的と必要性

記載内容が不十分であれば、どの様なシステムをユーザは望んているのかのイメージが掴みにくいため、ンステム構築する際に支障を来たす。

（3）説明

プロトタイプ段階では、完全な要求仕様が存在することは少なく、ラフな機能要件がまとめられている程度である。しかし、プロトタイピングの進展と共に、次第にシステムの目的、及び内容が詳細化、明確化されるものとする。その終盤には、機能・性能を含む完全な要求仕様書が作成されるへきである。そしてそれに基づき実運用ンステムが開発されることとなる。

要求仕様の種類には、開発フェーズに沿って、図3 1 4-2 に示す5種類が存在し、各要求仕様書は、各開発フェーズの最初の作業として作成するものとする。詳細な内容については、2 6 2節を参照のこと。

企画提案書（第0版要求仕様書）	（調査・計画フェーズ）
第Ⅰ版要求仕様書	（実験型プロトタイピングフェーズ）
第Ⅱ版要求仕様書	（発展型プロトタイピングフェース）
実用版要求仕様書	（実用システム開発フェース）
変更要求仕様書	（運用・洗練化フェーズ）

図3 1.4-2 要求仕様書の種類

なお要求仕様の内容に対する評価基準は、以下の通りてある。

表3 1 4 1-1　要求仕様の評価基準

評価基準	説　　明
正当性(correctness)	要求仕様が期待と合致している度合。
完全性(completeness)	要求仕様の記述に誤りや脱落がないこと。
一貫性(consistency)	要求仕様全体を通して矛盾のないこと。
伝達性(communicability)	要求仕様を誤りなく読者に理解させる度合。
連絡性(traceability)	関連する文書内の各要素との照合が容易に出る度合。
実現可能性 (implementability)	要求仕様に合致する製品を製作し、実現できる度合。

(4) 点検項目(C)・測定項目(M)

(C01)[正当性] 要求仕様の記述内容はユーザの期待と合致していること。
- 操作性、応答性、データ入力の簡便性、出力の視覚化、説明機能
- 推論時間、解の正解率、網羅性、最適性
- 他ノステムとのインタフェースなどの統合性

(C02)[完全性] 要求仕様の記述に誤りや記述漏れ、脱落がないこと。
(C03)[一貫性] 要求仕様全体を通して矛盾がないこと。
(C04)[伝達性] 要求仕様を誤りなく読者に理解させることが出来ること。
(C05)[連絡性] 関連する文書内の各要素との照合が容易に出来ること。
(C06)[実現可能性] 要求仕様に合致する製品を製作し、実現できること。

3142 要求仕様に対する達成度評価

（1）定義

エキスパートシステムが要求仕様で規定した要件をどの程度達成しているのかを評価することである。

（2）目的と必要性

要求仕様は、ユーザニーズを規定したものである。要求仕様に規定されている要件を達成していないシステムは当然ユーザの満足度が得られず、実用化されない可能性が高い。

（3）説明

要求仕様に対する達成度評価は、プロジェクト要件に対しては企画提案書（第0版要求仕様書）、機能・性能要件に対しては第Ⅱ版要求仕様書、実用版要求仕様書を中心に達成度評価を行うことになる。

（4）点検項目(C)・測定項目(M)

《実験型プロトタイピング段階》

(C01)［第Ⅰ版要求仕様書］第Ⅰ版要求仕様書の内容が、達成されているか否かを検討すること。

下記の項目の要求が達成されているかどうかを検討する。

- システム化の目的と範囲
- 機能要件（要求される機能、想定したデータの範囲、出力結果など）

《発展型プロトタイピング段階》

(C02)［第Ⅱ版要求仕様書］第Ⅱ版要求仕様書の内容が、達成されているか否かを検討すること。

下記の項目の要求が達成されているかどうかを検討する。

- システム化の目的と範囲
- 機能要件（要求される機能、想定したデータの範囲、出力結果など）
- 性能要件（応答時間）
- ユーザ要件（ユーザインタフェース、導入効果）
- 環境要件（資源、ハードウェアの制約など）

《実用システム開発段階》

(C03)［実用版要求仕様書］実用版要求仕様書の内容が、達成されているか否かを検討すること。

特に、下記の項目の要求が達成されているかどうかを検討する。

- システム化の目的と範囲
- プロジェクト要件（開発費、納期、要員）
- 機能要件（要求される機能、想定したデータの範囲、出力結果など）
- 品質・性能要件（応答時間、正確性、保守性、使用容易性）
- ユーザ要件（ユーザインタフェース、導入効果）
- 環境要件（資源、ハードウェアの制約など）

《運用・洗練化段階》

(C04)［計画と実際の開発とのギャップ］実運用が開始された頃に、当初計画した企画提案書の記述内容通りに開発が進められたのかどうかを検討すること。

計画と実際の開発とのギャップがどのくらいであるのかを把握する。

(C05)［変更要求仕様書］変更要求仕様書の内容が、達成されているか否かを検討すること。

《開発／保守方法・技法・ツールに関する評価》

3 1.4.3 要求仕様設定手段の評価

（１）定義

要求仕様を設定する手段が適切であるのか否かを評価することてある。

（２）目的と必要性

要求仕様を取りまとめる前に、記述する方法や手段を充分検討しておかないと、ユーザの要求するニーズを的確に表現できなかったり、要求仕様を固める際の作業効率に影響を及ぼす。

（３）説明

要求仕様を作成する際、専門家の主観に依存したトメインが対象となるため、システム分析の手法・技法をそのまま適用できるわけではなく、観点の依存性、異なる観点がある場合その関連性を考慮する必要がある。

要求仕様を設定するための方法は以下の通りである。

①ニーズ抽出法
- ５Ｗ１Ｈ法

②対象の構造分析をする方法
- ブロックモデル
- 階層構造モデル

③要求分析をするためのツール（例）
- ＩＳＭ　(Interpretive Structural Modeling)　[Warfield 1974]
- ＴＵＰＰＳ(Tool, User and Project-Planner System)[松村 1983]
- Ｃ－ＮＡＰ(Customer-Needs Analysis Procedures)　[富士通 1986]
- ＰＰＤＳ　(Planning Procedure to Develop Systems)[中尾 1980]

④要求モデル　[松本 1989]
- 有限状態機械
- データフローモデル
- 刺激応答パスモデル
- 機能階層モデル
- 並行プロセスモデル
- データ指向モデル

（Ｅ－Ｒモデル、オブジェクトモデル、関数的仕様、述語論理的仕様、代数的仕様、公理的仕様、操作的仕様）

（４）点検項目(C)・測定項目(M)

(C01)［ユーザのニーズ］用いた要求仕様記述の方法により、的確にユーザのニーズを表現できたかを評価すること。

(C02)［プロトタイプの利用］要求仕様の設定にプロトタイプを有効に利用すること。

(C03)［評価の役割分担］最初要求仕様の作成者自身が達成度の評価をし、さらに第３者がその内容をレビューすること。

3.1.5 知識ベースの保守性評価

解説の該当節　　▨印
3.1.1 知識ベースの妥当性評価
3.1.2 問題解決・推論方式の妥当性評価
3.1.3 結果の妥当性評価
3.1.4 要求仕様に対する達成度評価
3.1.5 知識ベースの保守性評価 3.1.5.1 変更箇所発見の容易性評価 3.1.5.2 知識ベースの理解性評価 3.1.5.3 変更箇所の局所性評価 3.1.5.4 知識ベースの統一性評価 3.1.5.5 品質安定性の確認方法の確認 3.1.5.6 保守マニュアルの整備状況の確認 3.1.5.7 保守用ツールの整備状況の確認
3.1.6 ヒューマンインタフェースの妥当性評価
3.1.7 システムの総合評価

（１）定　義

誤りの発見や利用者の要求の変化の時点のみならず、機能・性能の一層の向上のために、実稼働後も通常のソフトウェア以上に保守の頻度は高い。またエキスパートシステムにおける保守は、知識ベースが中心となる。知識ベースの保守性評価とは、システムが知識ベースの保守作業の遂行にどの程度適しているかを評価することである。

（２）目的と必要性

エキスパートシステムでは、設計段階で要求機能を厳密に決定することが困難であるため、誤りの発見や利用者の要求の変化の時点のみならず、機能・性能の一層の向上のために、実稼働後も通常のソフトウェア以上に保守の頻度は高い。そこで、知識ベースの保守性はエキスパートシステムの重要な評価項目である。また、この評価は、システムの改良や次期システムの開発の指針を得るためにも必要である。

（３）説明

知識ベースの保守が従来ソフトウェアと異なる点は、対象分野の専門家が作業に参加することである。そこで、専門家による保守を円滑化することが知識ベースに特有の問題となる。

保守の手順としては、図3.1.5-1に示すように、まずエキスパートシステムの変更をしたい機能あるいは強化したい性能に対して、知識ベース中の変更

すべき部分を見いだすことから始まる。つぎに、その部分を理解した後、必要な拡張、修正を行い、テストを行う。さらに、保守に伴って予期せぬ誤りが作り込まれていないこと（品質の安定性）を確保するために、システム全体の連動テストを行う場合もある。その過程で、各種のマニュアル、ツールを利用する。

そこで、評価項目として、変更箇所発見の容易性、理解性、修正の容易性（すなわち、変更箇所の局所性）、品質の安定性の確認方法、保守マニュアル及びツールの整備状況を取り上げた。

以上は、保守担当者からみた場合に必要となる評価項目であるが、一方、知識ベースのモジュラリティ、統一性（一貫した方針で作られていること）など、内部構造の観点からの評価項目も必要である。

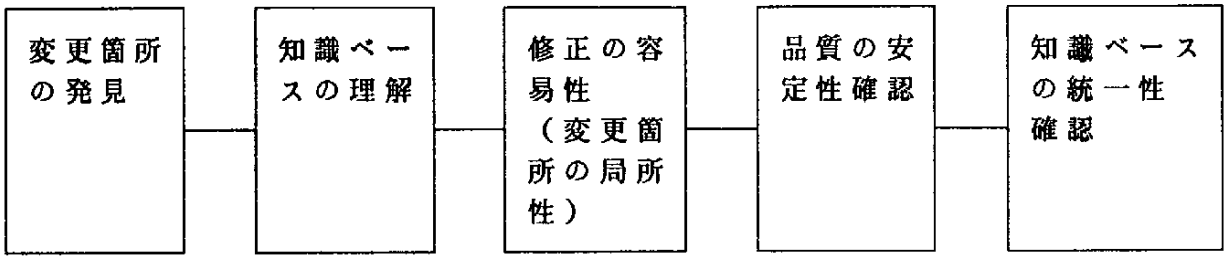


図3 1.5-1 保守の手順

《プロダクトに関する評価》

3.1.5.1 変更箇所発見の容易性評価

（1）定　義

変更したい機能に対して、知識ベース中の変更箇所がどの程度容易に見いだせるか、あるいは、知識ベースの一ヶ所を変更したときに、関連する変更箇所をどの程度容易に見い出せるかを評価することである。

（2）目的と必要性

変更箇所発見の容易性は、保守に伴う誤りの作り込みを防止するため、また、保守の効率確保のために重要である。評価結果は、知識のモジュール分割の改良に反映される。

（3）説　明

変更箇所発見の容易性は、エキスパートシステムの外部機能と内部機能の対応、知識モジュール間の独立性に依存する。以下の留意点を考慮して評価する。

①モジュール分割

②追跡可能性（トレーサビリティ）

③ツールによる環境整備

④ドキュメントの整備

（4）点検項目(C)・測定項目(M)

(C01)［モジュール分割］モジュール分割が、システムの外部機能と対応していること。

(C02)［推論処理に対応したモジュール化］モジュール分割が、一まとまりの推論処理と対応していること。

(C03)［モジュラリティの向上］従来言語で記述した部分は、モジュラリティの向上が図られていること（例えば、情報隠蔽、同一機能を一ヶ所にまとめることなど）。

(C04)［トレース機能1］推論結果はどの様にしてまた、どのような条件下で生成されたのかを容易にトレースできる機能があること。

(C05)［ツール、機能の整備］ルールエディタ、用語辞書、クロスレファレンス等が整備されていること。

(C06)［ドキュメントの整備］トキュメントは整備されていること。

(C07)［名称の分かりやすさ］ルール名称やフレーム名称等が分かりやすくつけられていること。

(C08)［トレース機能2］オブジェクト指定のルールトレース機能があること。

(C09)［制御知識と結果の関係］知識ベース内の制御知識と出力結果との因果関係が明確であること。

3.1.5.2 知識ベースの理解性評価

（１）定　義

保守担当者（専門家、ＫＥ、ユーザなど）が、知識ベースの内容をどの程度容易に理解できるかを評価することである。

（２）目的と必要性

理想的には、知識ベースの内容を理解しなくても知識が追加でき、修正部分のみ理解すれば修正できるのが望ましい。しかしながら、そのためには、知識の冗長性や矛盾をシステムが自動的に検出、修正する必要があり、現状のエキスパートシステム技術ではこの理想は実現できない。従って、保守に当たっては、保守部分を含めた一定範囲の知識を理解する必要がある。そこで、知識ベースの理解性を評価項目として取り上げる。評価結果は、システムの改訂や次期システムの開発時に、対象世界のモデル、知識のモジュール分割、記述方法の改良に反映される。

（３）説　明

前述のように、知識ベースの保守には、ＫＥの他、対象分野の専門家が参加する。特に、知識ベースが複数の分野の知識を含む場合には、異なる分野の複数の専門家が参加する。

以下の留意点を考慮して評価する。

①保守のしやすい構築法
②対象世界のモデルの素直さ、分かりやすさ
③知識表現の分かりやすさ
④保守ドキュメントの整備

（４）点検項目(C)・測定項目(M)

(C01)［専門家による保守］専門家による保守を想定したモジュールは、プログラムの知識がなくても理解できるようになっていること。

(C02)［モデルの素直さ］対象世界のモデルが素直で、分かりやすいこと。

(C03)［記述レベル1］知識ベースが、機能と構造の対応、知識表現、従来型言語で記述した部分の記述レベルの高さの観点から分かりやすいこと。

(C04)［記述レベル2］専門家が意味を理解しやすい、記述レベルが統一できる高レベルの言語で記述してあること。

(C05)［見通しのよい知識構造］知識ベースのモジュール化、分散協調化を図り、推論の見通しをよくするよう工夫すること。例えば、ルール内変数の採用、フレーム表現にし直す、手続き的に記述するようにする。

(C06)［技術移転］開発段階から専門家も参画し、技術移転がスムースに行われるように心がけること。

(M01)ルール数／メタルール（制御のためのルール）数の測定

3.1.5.3 変更箇所の局所性評価

（１）定　義

エキスパートンステムの性能、対象範囲の拡張や修正に対して、知識ベース中の変更箇所がどの程度限定されるかを評価することである。

（２）目的と必要性

変更箇所の局所性は、保守の効率を確保するため、及び保守に伴う誤りの作り込みを防止するために重要である。また、専門家による保守を円滑化するためには、変更箇所が専門家の担当範囲を越えて波及しないこと、例えば従来言語で記述されたプログラム部分に及ばないことが重要である。そこでこの性質を評価項目として取り上げた。評価結果は、知識のモジュール分割の改良に反映される。

（３）説　明

専門家による保守を円滑化することに関連し、以下の留意点があげられる。

①保守担当者を明確に想定した上での開発

②モジュール間のインタフェースの明確化

③変更の少ない知識と多い知識のきり分け

さらに、ソフトウェア全般の評価項目として、モジュラリティに関するものなどがある。これらは、３１５１節の変更箇所発見の容易性で述べた①～②と同様である。

（４）点検項目(C)・測定項目(M)

(C01)［モジュール間のインタフェース］異なるモジュール間のインタフェースが明確に定義されていること。

(C02)［変更の少ない知識と多い知識］変更の少ない知識と多い知識が切り分けられていること。

知識ベースには教科書的知識のように変更の少ないものと、経験的知識のように比較的変更の多いものがある。そこで変更範囲を局所化するためには、変更の多い部分とをきり分けておくことが有効である。

(C03)［記述の重複］記述の重複が最小限になるように適切な階層化が行われていること。

(C04)［同一の項目］同一の項目が何箇所に重複して記述されていないこと。

(C05)［メタルール］メタルールの利用を極小化すること。

(C06)［変更可能な範囲］保守対象の知識と変更可能な範囲を明確にし、エラーが起きないように設計すること。

(M01)問題あたりの修正ルール数の測定

3.1.5.4 知識ベースの統一性評価

（１）定義

知識ベースがどの程度一貫した方針で作られているかを評価することである。

（２）目的と必要性

保守が長期にわたる場合や保守の担当者が複数である場合は、知識の競合や矛盾を避けることは困難である。しかし知識ベースが一貫した方針で作られていれば、その方針に従って保守を行うことにより、誤りの発生を減らすことができる。また担当者のレベルがまちまちである場合にも、ある程度知識ベースの質を保つことができる。そこで、知識ベースの統一性を評価項目としてとりあげた。

本評価結果は、システムの改訂や次期システムの開発時に、対象世界のモデルや知識のモジュール分割に反映される。

（３）説明

（知識ベースだけでなく、何事に関してもいえることだが）統一性は、方針が明確かつ一貫しているかどうか、方針が周知徹底されているかとうかに依存する。以下に関連する留意点をあげる。

①対象世界のモデルの明確化、一貫性

②モデルの忠実性、一貫性

（４）点検項目(C)・測定項目(M)

(C01)［対象のモデルの明確性］対象のモデルが明確かつ一貫していること。

(C02)［モジュール分割の一貫性］モジュール分割の観点が一貫していること。

(C03)［知識表現の一貫性］知識の種類に対する知識表現が一貫していること。

(C04)［記述レベル］専門家が意味を理解しやすい、記述レベルが統一できる高レベルの言語で記述してあること。

(C05)［記述の一貫性］記述（名前やデータ構造の選択）が一貫していること。

《開発／保守方法・技法・ノールに関する評価》

3155 品質安定性の確認方法の確認

（1）定　義

保守に伴って予期せぬ誤りが作り込まれていないことを検証するための方法が確立されているかどうかを評価することである。

（2）目的と必要性

エキスパートンステムでは、断片的な知識が実行時に連動するため、知識の修正に伴う副作用を発見、修正することが、従来ソフトウェアに比へ困難である。そこで品質の安定性を確認することが重要であり、そのための方法が確立されている必要がある。評価結果は、保守マニュアルに反映される。

（3）説明

保守前のンステムに対するテストを再度行うなどの方法が確立され、それが保守マニュアルなどに記載されているかどうかを評価する。

（4）点検項目(C)・測定項目(M)

(C01)［標準問題集の有無］標準問題集があること。

(C02)［動作確認用のサンプルデータの有無］動作確認用のサンプルデータが用意されていること。

(C03)［バージョン管理］バージョン管理がされていること。

(C04)［変更による影響伝搬］変更による副作用を含めた影響伝搬が確認できること。

(C05)［解の質低下］保守にともなって、解の質が低下しないこと。

(C06)［解の質向上］洗練化の対象となった問題に関して、解の質が向上すること。

3 1 5.6 保守マニュアルの整備状況の確認

（１）定　義

保守に必要なマニュアルが整備されているかどうかをを確認することである。

（２）目的と必要性

保守マニュアルの整備は、保守の効率確保、及び知識ベースの質の確保のために重要であるから、評価項目として取り上げた。評価結果は、保守マニュアルの拡充に反映される。

（３）説　明

以下の評価項目があげられる。

①必要な保守マニュアルが揃っているか（例えば、各種保守作業の手順マニュアル、ツールの操作マニュアルなど）。

②各マニュアルには必要事項が正確に記載されているか（保守環境、作業、操作、事後の確認、記録の方法など）。

③標準にそった一貫した記述になっているか。

④読者（保守担当者(専門家、ＫＥ、ユーザなど)）を明確に想定し、想定読者が理解できるような説明になっているか。

（４）点検項目(C)・測定項目(M)

(C01)［保守マニュアル］保守担当者にとり必要な保守マニュアルの内容が正確に記述されていること。

(C02)［文書化基準］文書化基準に沿って一貫した記述になっていること。

(C03)［保守担当者1］保守担当者を明確に想定し、その担当者に理解が出来るような説明となっていること。

(C04)［専門家による保守］専門家が行う保守では、マニュアルを見ずに作業できることが望ましい場合もあること。

(C05)［保守担当者2］保守担当者の立場により、記述内容のレベル、理解しやすさ等を考慮すること。

(C06)［保守履歴］保守履歴が管理されていること。

3.1.5.7 保守用ツールの整備状況の確認

（１）定　義

保守に必要なツール類が整備され、容易に使用できる状況になっているかどうかを確認することである。

（２）目的と必要性

保守用ツールの整備は、保守の効率確保、及び知識ベースの質の確保のために重要である。評価結果は、ツールの拡充に反映される。

（３）説　明

知識ベースの保守に特有のツールとしては、専門家用ツールがある。どのような専門家用ツールが望ましいかは一概に決められないが、一般的にはプログラムの知識がなくても操作でき、専門家が日頃親しんでいる表現（ワークシートや自然言語）や用語により知識を保守できるツールが望ましい。

専門家用ツールの種類としては、以下のようなものがある。

①分野別知識エディタ（ワークシートや自然言語による入力、セマンテックチェックなどの特徴がある）

②説明ツール（推論過程の説明など）

③解析ツール（ルールやフレームの関連表示）

④テストツール（ルールのステップ実行、テストデータの管理、再実行など）

（４）点検項目(C)・測定項目(M)

(C01)［ツールの整備］保守の状況に応じて、知識エディタ、説明ツール、解析ツール、テストツール等を整備すること。

- 知識エディタ（セマンテックなチェックが可能）
- 説明ツール（推論過程の説明）
- 解析ツール（ルールやフレームの関連表示）
- テストツール（ルールのステップ実行、テストデータの管理、再実行など）

(C02)［専門家による保守］専門家が保守をする場合は、特に、プログラムの知識がなくとも操作ができ、専門家が日頃慣れ親しんでいる表現（ワークシート、自然言語など）や用語を用いて保守ができること。

3.1.6 ヒューマンインタフェースの妥当性評価

解説の該当節　　▨▨▨▨印
3.1.1 知識ベースの妥当性評価
3 1.2 問題解決・推論方式の妥当性評価
3 1 3 結果の妥当性評価
3 1.4 要求仕様に対する達成度評価
3.1.5 知識ベースの保守性評価
3.1.6 ヒューマンインタフェースの妥当性評価 3.1.6.1 利用時のインタフェース評価 3.1.6.2 保守時のインタフェース評価
3.1.7 システムの総合評価

（１）定義

エキスパートシステムのヒューマンインタフェースが、利用時、保守時ともに使いやすくできているか否かを評価することである。

（２）目的と必要性

ヒューマンインタフェースは、エキスパートシステムやＥＳ構築ツールの顔であり、提供されている機能がいかにすくれていても、あるいは、性能が良くても、ヒューマンインタフェースが良くないと使われない運命をたどる可能性が非常に大きいので、ヒューマンインタフェースの評価は極めて重要である。

（３）説明

利用時インタフェース、保守時インタフェースの違いを以下に述べる。

① 利用時インタフェースの範囲・利用対象

- 開発されたエキスパートシステムを一般利用者が利用する局面
- 適用業務の利用者（専門家または非専門家）が利用対象

② 保守時インタフェースの範囲・利用対象

- 知識ベース保守時は、適用業務の利用者、ナレッジエンジニアが利用対象

《プロダクトに関する評価》

3.1.6.1　利用時のインタフェース評価

（１）定義

開発されたエキスパートシステムの利用時インタフェースの良さを評価することである。

（２）目的と必要性

利用時インタフェースが悪いとエキスパートシステムの実用性が失われる可能性が高く、その評価は重要である。

（３）説明

エキスパートシステムのヒューマンインタフェースには、一般の情報処理システムと同様の面とエキスパートシステム特有の面とがある。以下に主要なヒューマンインタフェースの評価項目を示すが、このうち「説明機能」はエキスパートシステム特有の項目である。

以下の留意点を考慮して評価する。

①表示機能と操作性
②ガイダンス機能
③エラー処理
④自然言語機能
⑤処理速度
⑥説明機能

（４）点検項目(C)・測定項目(M)

《表示機能と操作性》

(C01)［結果の表示］結果の表示や判断・応答の入力補助情報の表示において適切な表現方法が用いられていること。

- 文脈（Context）の表示方法・順序
- 情報の内容に応じた表現形式（図、グラフ、リスト、グラフィック、色）
- 情報の重要度に応じた表現
- 関連情報への容易なアクセス
- 標準への遵守性（情報の正しい解釈を助ける）

(C02)［入力の操作性］入力の操作性が良いこと。

- 専門家が通常行っている操作と類似していること
- 標準への遵守性（操作の誤りをなくす）
- 入力機器・媒体の選択
- 長時間使用しても身体的な苦痛が少ない

(C03)［利用者インタフェース］マウスやマルチウィントウなどの利用者インタフェースは、開発工数が増えるわりに利用者に喜ばれないこともあるので注意すること。

《ガイダンス機能》

(C04)［ＨＥＬＰ機能］ＨＥＬＰ機能の充実とその操作性がよいこと。

- 文脈による必要最小限のＨＥＬＰ
- 利用者主導の能動的なＨＥＬＰ
- 操作誤り等による受動的なＨＥＬＰ

(C05)［文脈の表示］明確な文脈の表示（明示／暗示）であること。
- 質問の意味
- 入出力情報の意味
- 次に何をすべきかが理解できる

(C06)［習熟度］利用者の習熟度に応じたガイダンスであること。
- 習熟の速さ・正確さ
- 習熟度に応じた表現

《エラー処理》

(C07)［エラー処理］利用者の誤りによって生じたエラーに対する処置が出来ていること。
- 利用者の操作誤りによるエラー処理
- エラーの認知機能
- 回復機能(Undo)
- 再開始機能(Redo)

《自然言語機能》

(C08)［曖昧さの表現］曖昧さや不確定性の表現方法として適切であること。

(C09)［説明機能］より理解しやすい結果の説明機能として適切であること。

(C10)［使用する局面］質問の生成、結果の表示、応答入力、ガイダンス／説明機能の実現が適切であること。

《処理速度》

(C11)全体の処理時間

(C12)応答時間

《説明機能》

(C13)ＷＨＹ機能
- 何故この様な質問をするのかを説明する。
- 現在の文脈を理解させる。
- 正しい応答入力を促す。

(C14)ＨＯＷ機能
- どの様な推論過程でそのような結論に至ったのか。
- 結果の妥当性を説明する。
- 結果の正しい解釈を促す。

(C15)ＷＨＡＴ機能
- より詳しい質問の内容の表示。
- 利用者の習熟度に応じた質問方法。

(C16)［利用環境］運用者（利用者）の利用環境を考慮したヒューマンインタフェースとなっていること。

(C17)［対話性］冗長な質問を出さないこと、利用者主導の対話制御が可能であること。

(C18)［取扱の範囲］取り扱い範囲の限界を認識し適切に応答できること。

(C19)［利用者の目的］［利用者の知識］［利用者の習熟度］利用者の目的、知識、習熟度に応じた評価が必要であること。

3 1 6.2 保守時のインタフェース評価

（１）定義

エキスパートシステムの知識ベースの保守を行う場合のインタフェースの良さを確認することである。

（２）目的と必要性

エキスパートンステムにおける知識ベースの保守は運用段階に入っても継続され、また開発者のみならず専門家を始めとする利用者自身によって保守が行われるケースも多い。保守時のヒューマンインタフェースは保守作業の生産性と質を向上させるため極めて重要である。

（３）説明

知識ベースの保守では、その目的の支援ツールを用いる場合と、エキスパートシステム自体が持つその機能を用いる場合がある。いずれの場合もヒューマンインタフェース上の評価の留意点は以下の通りである。

①知識ベースの可視性
②変更すべき箇所の同定の容易性
③矛盾や一貫性等を検査する機能
⑤既存のテストデータによる回帰テストの機能

なお、表示機能、操作性、ガイダンス機能など一般的なヒューマンインタフェースに関しては「利用時のインタフェース評価」と同様である。また保守を担当する人が、開発者か利用者により、それぞれの専門性やレベルに応じたインタフェースであることが必要不可欠である。

（４）点検項目(C)・測定項目(M)

(C01)［保守担当のレベル］保守を担当する人が、開発者か利用者により、それぞれの専門性やレベルに応じたインタフェースであること。

(C02)［ビジュアル化1］推論途中をビジュアルにトレースできるインタフェースがあること。

(C03)［変更による影響］知識の部分変更を行ったとき、他にどんな影響を与えるかを示してくれるインタフェースがあること。

(C04)［モジュラリティ］保守時の誤り防止に知識ベースのモジュラリティを利用すること。

(C05)［ビジュアル化2］知識ベースの可視性を高める機能を持つこと。

(C06)［変更箇所の同定］変更すべき箇所の同定を容易にする機能を持つこと。

(C07)［内部表現への変換］専門家の知識表現によって入力した知識をシステム固有の表現に変換する機能を持つこと。

(C08)［矛盾の検査］［一貫性の検査］保守時の知識ベースの品質低下が起こらぬよう、矛盾や一貫性等を検査する機能を持つこと。

(C09)［回帰テスト］既存のテストデータによる回帰テストが行える機能を持つこと。

3.1.7 システムの総合評価

解説の該当節　　▨印
3.1.1 知識ベースの妥当性評価
3.1.2 問題解決・推論方式の妥当性評価
3.1 3 結果の妥当性評価
3.1.4 要求仕様に対する達成度評価
3.1.5 知識ベースの保守性評価
3.1.6 ヒューマンインタフェースの妥当性評価
3.1.7 システムの総合評価 3.1.7.1 テーマ選定の妥当性評価 3.1.7.2 利用者の満足度評価 3.1.7.3 統合性の評価 3.1.7.4 成長可能性の評価 3.1.7.5 開発／保守用ドキュメントの評価 3.1.7.6 開発方法の適切さの確認 3.1.7.7 ツール・言語選択の適切さの確認 3.1.7.8 システム開発の生産性評価 3.1.7.9 システムのコストパフォーマンス評価 3.1.7.10セキュリティ対策の評価

（１）定義

エキスパートシステムが情報処理システムとして総合的にみて満足できるものかどうかを評価することである。

（２）目的と必要性

情報処理システムの評価は、部分やある視点での評価に加えて、システム全体としてのバランスや機能間の統合性など総合的な観点からの評価が必要である。特に、エキスパートシステムでは知識処理的な部分とネットワークやデータベースを始め従来型技術による部分とが統合されている可能性が高く、総合性評価の必要性は極めて高い。

《プロダクトに関する評価》

3.1.7 1 テーマ選定の妥当性評価

（１）定義

エキスパートシステムとしてテーマの選定が妥当であるのか否かを評価することである。

（２）目的と必要性

従来システムのアプローチでも充分対処できる内容のシステムを、エキスパートシステムとして構築しても充分な効果が得られぬことが多い。

テーマ選定の良否はエキスパートシステム評価の基本要件である。

（３）説明

実用化を狙うエキスパートシステムでは、「エキスパートシステム化へのニーズの高さ」、「エキスパートシステム向きの問題」、「解の完全度の要求度」、「プロトタイピングへの適用性」、「知識獲得の成功の可能性」等を考慮する必要がある。

①エキスパートシステム化へのニーズの高さ

- ユーザがエキスパートシステムへの投資効果が高いと感じていること。
- 実際に使われることを想定としていること。
- システム化の対象範囲や制限条件を現実的に把握が可能であること。

②エキスパートシステム向きの問題

- 専門家の経験的な知識を記号処理により問題解決できる問題であること。
- 解法そのものが厳密に定義しがたい、あるいは解法は分かっているが複雑に入り組んでおり、アルゴリズムやデータの構造の設計が難しい問題であること（組合せの爆発が生じる問題）。
- 従来のプログラム手法で問題解決は出来るが、エキスパートシステムのプロトタイピング的アプローチを採用した開発を行いたい場合であること（開発の方法論として、プロトタイピング手法を採用）。
- 問題の領域が安定しており、開発運用中に大きな変化がないこと。
- 従来のプログラムであると、条件が頻繁に変わるためその都度の作り直しはかなわない問題であること（変更への対応と迅速化）。

③解の完全度の要求度

下記の何れかであれば推奨する。

- 解析型問題で１００％の正解率を要求しない。
- 合成型問題で最適解を要求しない。
- 最後は人間が判断する。

④プロトタイピングへの適用性

- システムは逐次的に成長発展が可能であること。
- 開発に時間とコストをかけられる問題であり、当初は実験システムとしての性格を持つ問題であること。

⑤知識獲得の成功の可能性

- 真の専門家が存在する問題であること。
- 専門家の絶大な協力が得られること（時間的、心情的にも）。
- 専門家は知識をより明確に話せること。

（４）点検項目(C)・測定項目(M)

《ニーズ性》

(C01)［ニーズの強さ］［ニーズの分析］システムに対するニーズの強さを表３１７-1に示すような観点から充分整理・検討を行うこと。

表３１７-1　ニーズの分析

ニーズの分類	ニーズの詳細化
作業効率の向上	生産性の向上 省力化 専門家の減少 作業のやり直しの減少
専門家の作業代行	自動化 業務の大幅な拡大
専門家の作業支援	専門家の負荷軽減 必要情報の入手にかかる時間の短縮化
作業の質的改善・向上	業務の高品質化 業務の均質化 誤りの減少・防止 戦略情報システム化
教育	初心者の教育 専門家の育成 熟練技術の伝承 知識の蓄積・標準化（普及、共有、継承）
システム化によるメリット	知識の整理・体系化 ノウハウの知識ベース化 保守や拡張の容易化 エントユーザによる操作の容易化
その他	サービスに対する付加価値の向上 新しいビジネスチャンス 商品の差別化の手段 知識工学／ｴｷｽﾊﾟｰﾄｼｽﾃﾑの技術修得、評価

《適切な問題》
(C02)［明示的な知識］経験的知識やノウハウが記述できる問題であること。
(C03)［アルゴリズムでは困難］従来のアルゴリズムやデータ構造では設計が難しい問題であること。
(C04)［試行錯誤性］試行錯誤の過程を含む問題であること。
(C05)［組合せの膨大］組合せが膨大な問題であること。
(C06)［論理の単純性］ルールは多く複雑だが論理は単純な問題であること。
(C07)［データの利用］過去のデータや結果が一部利用可能な問題であること。
《専門家の存在》
(C08)［専門家の存在］対象問題の専門家が存在し、その協力が得られること（時間的、心情的にも）。
(C09)［円滑なコミュニケーション］専門家との円滑なコミュニケーションが可能であること。
(C10)［インタビューの時間］インタビューの時間が充分に確保できること。
《結果の妥当性》
(C11)［正解率］必ずしも１００％の正解率を要求しないこと。
(C12)［最適解］必ずしも最適解を要求しないこと。
(C13)［解の判断］結果の妥当性が開発グループで判断できること。
《プロトタイピング》
(C14)［ﾌﾟﾛﾄﾀｲﾋﾟﾝｸﾞｱﾌﾟﾛｰﾁ］開発の方法論として、プロトタイピング手法が採用できること。
(C15)［変更への対応と迅速化］従来のプログラムであると、条件が頻繁に変わるためその都度の作り直しはかなわない問題であること。
(C16)［システムの成長可能性］システムは逐次的に成長発展が可能であること。
(C17)［開発の時間とコスト］開発に時間とコストがかけられる問題で、当初は実験システムとしての性格を持つ問題であること。

3.1.7.2 利用者の満足度評価

（１）定義

利用者が対象とするシステムにどの程度満足しているかを評価することである。

（２）目的と必要性

実用システムとして使われるか否かを見極めるための最重要評価項目は、利用者の満足度であり、評価が必須である。

（３）説明

利用者の満足度を高めるには、例えば下記の項目が実現されている必要がある。

- 欲しい機能が実現され、提供されている。
- 操作が簡単である。
- レスポンスが速い。
- 入力データの準備に時間がかからない。
- 必要最小限のデータを入力すれば済む。
- 出力結果が視覚化されている。
- 出力結果の信頼性が高い。
- 説明機能があり、分かりやすい。
- 学習機能があり、操作性が向上する。
- 再開機能があり、利用を中断することが出来る。
- システムを使うことによって作業の生産性が向上する。
- システムを使うことによって作業の質が向上する。
- 営業において、システムが差別化の手段となる。
- ユーザの改善要望を聞き、改善・機能拡張がタイムリーに行われている。
- 教育プログラムが充実し、随時実施されている。
- ユーザマニュアルが充実し、分かりやすい。

以上のような項目を、満足度を決定付ける重要な因子、例えば、操作性、機能性、解の信頼性、戦略性などと対応づけ、レーダチャートやフェース分析（Face Analysis）の形式で表現して、ユーザの満足度を総合的に評価する方法がある。

（→本評価項目は、「要求仕様に対する達成度評価」、「利用時のインタフェース評価」、「システムのコストパフォーマンス評価」と関連する。）

（４）点検項目(C)・測定項目(M)

(C01)［現場に近い環境］出来る限り現場に近い環境で評価すること。

(C02)［要求仕様］予め要求仕様に利用者の満足度を評価する項目、目標を記述しておくこと。

《導入効果》

(C03)［導入効果、利点］エキスパートシステム導入により得られた効果、利点を表３１７-1の項目でチェックすること。

(C04)［導入弊害］エキスパートシステム導入により発生した以下のような弊害を調べること。

- 職場モラルの低下、拒否反応
- 定型の結果しか得られない
- 責任の所在が不明確
- 技術のブラックボックス化
- 慣れるのに手間、時間がかかる
- 専門家のスキル向上の停滞
- 新しい専門家を育成しがたい

《使用容易性》

(C05)［必要機能］欲しい機能が実現され、提供されていること。

(C06)［操作の簡便性］操作が簡単であること。

(C07)［レスポンス］レスポンスが速いこと。

(C08)［入力データ］入力データの準備に時間がかからないこと。

(C09)［必要な入力データ量］［最小限のデータ］必要最小限のデータを入力すれば済むようにすること。

(C10)［視覚化］出力結果が視覚化されていること。

(C11)［出力結果の正当性］出力結果の正当性が高いこと。

(C12)［説明機能］説明機能があり、分かりやすいこと。

(C13)［学習機能］学習機能があり、操作性が向上すること。

(C14)［再開機能］再開機能があり、利用を中断することが出来ること。

《保守性》

(C15)［改善要望］ユーザの改善要望を聞き、改善・機能拡張がタイムリーに行われること。

(C16)［教育プログラム］教育プログラムが充実し、随時実施されていること。

(C17)［ユーザマニュアル］ユーザマニュアルが充実し、分かりやすいこと。

3.1.7.3 統合性の評価

（１）定義

従来型技術による機能との連携や統合性の良さを評価することである。

（２）目的と必要性

知識処理的機能と従来型機能との統合は、実用的なエキスパートシステムでの必須条件であるが、両者の最適な分担や効率上の負荷を最小限におさえるなど課題も多く、その統合性に対する評価は必要不可欠である。

（３）説明

多くの実用段階でのエキスパートシステムは、スタンドアロンのンステムとしてではなく、既存の機能や技術と連携して、双方の良い点を生かして実用性の高いシステムにまとめ上げている。例えば、通信機能、データベース、統計処理、数値計算、画像処理などの従来技術による機能との統合が行われることが多い。

これらの統合化に際しては、既存機能にできるだけ手を加えたくないという配慮が働くことが多く、その結果、効率やメモリスペース上の負荷が過度になる、部分的機能の重複がおこるなど、望ましくない結果となる可能性がある。特にデータベースと知識ベースの統合は両者のデータ構造がかなり異なるため、一般にその間のインタフェースとして変換機能が必要となり、その品質が問題となる。

既存機能との統合化は、次の４つの形態に大きく分けられる。それぞれの形態毎にその連携がよいのかどうかを評価する。

①エキスパートシステムから既存機能を呼び出す形式

例）各種パッケージのＣＡＬＬ
　　データベースの利用
　　画像制御ルーチンの利用
　　通信機能の利用

②既存システムのフロントエンドとしてエキスパートンステムを実現する形式
（この場合、ユーザに対し知的なユーザインタフェースを期待することが出来る。）

例）データベースの知的検索
　　知的ユーザインタフェース機能

③既存システムの一部をエキスパートシステム化して、付加価値を高めた形式

例）プロセス制御の監視機能部分

④既存機能との協調型の形式

（４）点検項目(C)・測定項目(M)

(C01)［統合による利点］エキスパートシステムと他システムとを統合したことによる利点を明確にしておくこと。

(C02)［既存機能または既存システムとの連続性］他ンステムとの連続性があること。

他システムとの統合化は、次の４つの形態を意識して評価する必要がある。

- エキスパートシステムから他システムを呼び出す形式
- 他システムのフロントエンドとしてエキスパートシステムを実現する形式
- 他システムの一部をエキスパートシステム化して、付加価値を高めた形式
- 他システムとの協調型の形式（既存機能と知識処理機能が別タスクで起動）

3.1 7 4　成長可能性の評価

（１）定義

エキスパートンステムの対象範囲の拡大や機能の向上など、システムの成長可能性を評価することである。

（２）目的と必要性

エキスパートシステムはプロトタイピングにより部分的な試作を積み重ねて対象範囲を拡大して行くが、その成長の可能性が明確でないと、開発の中断や、使用の中止などに至る可能性がある。これらの即断を避けるためにも、その成長の可能性を見極めることは重要である。

（３）説明

エキスパートシステムにおいては、一般に開発が完了すれば実用に耐えられるシステムが完成すると考えることが出来ず、運用中も知識が拡張されシステムが成長することを前提としなければならない。

従って、現時点でのエキスパートシステムの能力は、専門家と比較して幼稚であり劣っているとの評価結果が出された場合でも、エキスパートシステムの成長可能性を忘れてはならない。今後、知識を追加、修正し増大させて行くことにより、エキスパートシステムの判断能力なとを成長発展させる見込みがあるのかどうかがキーポイントである。

（４）点検項目(C)・測定項目(M)

(C01)［レビュー1］知識提供の専門家、開発担当者、意思決定するマネジャ、第３者の専門家等からなるレビュー会議において、エキスパートンステムの成長可能性について充分検討すること。

(C02)［レビュー2］特に、プロトタイピングンステムの段階において、エキスパートシステムの成長の可能性について充分レビューすること。

(C03)［総合判断］現時点でのエキスパートシステムの能力は専門家と比へて幼稚であり、劣っていると評価された場合でも、エキスパートンステムの成長の可能性を考慮して総合判断をすること。

(C04)［プロトタイプ］［期待度／失望度］プロトタイプを見せたときの反応について意見を聞くこと。

- 現状での期待度／失望度
- アプローチとしての期待度（今後、この方法で真面目に開発すると、今はおもちゃでも将来的には使いものになりそうだ、という直感的感覚）

(C05)［拡張の予定］知識ベースはどのあたりまで拡張を予定して設計してあるのかを確認すること。

- 推論エンジン、推論戦略知識、専門知識の分離（保守性）
- 知識修正、追加が容易に行われるエディタ環境の整備

(C06)［開発環境と実行環境］開発マシンと実行マシンとは同一である必要はないが、開発環境は実用化レベルへの発展を考慮にいれて準備されていること。

(C07)［改良の見込み］現状の性能・機能が充分でなくても、将来改良できる見込みがあること。

(M01)［資源の見積り］知識ベースの規模増大に対する必要資源の増加見積り（メモリー、計算コスト、知識獲得工数）を計算すること。

3.1.7.5 開発／保守用トキュメントの評価

（１）定義

開発用ドキュメント、保守用ドキュメントがどの程度完備され、その内容がどの程度充実しているのかを評価することである。

（２）目的と必要性

知識ベース、推論機構の正しさを調へる際、開発トキュメントをレヒューして確かめる必要があり、その評価は重要である。

（３）説明

ドキュメントの充実の必要性は、システムの状態により異なるが、少なくとも実用システムにおいてはその完備は必要不可欠であり、またトキュメントによるレビューも重視される。なお、各開発フェーズで作成されるトキュメントの内容は、２６２節を参照のこと。

（４）点検項目(C)・測定項目(M)

《ドキュメント全体》

(C01)［必要なドキュメント］必要なドキュメントが揃っていること（変更等に係わるドキュメントも含む）

(C02)［バージョンアップ1］ドキュメントはバージョンアップされていること。

(C03)［バージョンアップ2］運用フェーズになってもバージョンアップのルーチンは維持できること。

(C04)［ドキュメントの記載］トキュメントの記載は充分であること。

(C05)［ドキュメント形式］ドキュメント形式が標準にあっていること。

- 開発用ドキュメントは、文書化基準に準拠して作成されているか。
- 保守用ドキュメントは、文書化基準に準拠して作成されているか。

(C06)［ドキュメント内容］トキュメント内容は、矛盾がなく意味的に正しいこと。

(C07)［分かりやすさ］ドキュメントは、分かりやすくかつ誤解がないように書かれてあること。

(C08)［レビュー］トキュメントをレビューすることにより、そのエキスパートシステムの機能、性能の妥当性及び開発上の問題点等が充分指摘できること。

《ワークシート関連》

(C09)［網羅性］仕様書や設計書に書くへき事項が網羅的で、かつ適切てあること。

(C10)［体系化］ワークシート（ＷＳ）が体系化され、完備していること。

(C11)［記入項目1］ＷＳの記入すへき項目が必要充分で、かつ適切であること。

(C12)［記入項目2］ＷＳの記入項目が記入しやすいこと。

(C13)［記入例］ＷＳの記入例が豊富であること。

(C14)［チェックリスト］ＷＳに対するチェックリストが豊富であること。

(C15)［読み易さ］ＷＳが読み易く、レビューしやすいこと。

《ソフトウェアの機能》

(C16)［支援ツール］今後、マニュアルの整備にともない使いやすくするために、以下の機能を検討すること。

- オンラインマニュアル
- 多インデックス検索機能
- マルチメディア／ハイパーカード
- ＨＥＬＰ機能との一体化
- 知的チュートリアル

《開発／保守方法・技法・ツールに関する評価》

3.1.7.6 開発方法の適切さの確認

（１）定義

開発方法が適切であったのかどうかを確認することである。

（２）目的と必要性

システムの開発方法は、開発の生産性や成果物の品質に大きな影響を与えるため、その評価は重要である。

（３）説明

開発方法を規定する要素は次のようなものである。

- 開発工程の分割
- 開発方法論（技法）、及び利用ツール
- 各工程における作業内容
- 工程毎のドキュメントの種類と内容
- 各工程の開発管理（レビュー、完了の確認）方法

有効性が実証された開発方法を採用し、そこで規定された通りに作業を行えば、効率よくかつ高質な作業を実施できる可能性が高くなる。

開発方法の適切さに関しては、下記の点を評価すべきである。

①作業関連

- 工程分割の妥当性
- 開発方法論の適切性
- 各工程で行うべき作業内容の適切性
- 各工程で使用する技法、ツールの適切性と有効性

②管理関連

- 開発管理の基本方針
- 各工程の管理方法

③ドキュメント関連

- ドキュメント体系
- ドキュメントの記述基準　　• 各ドキュメントの記述内容

（４）点検項目(C)・測定項目(M)

(C01)［体系的な開発方法］体系的な開発方法であること。

(C02)［網羅的な開発方法］全ライフサイクルを網羅した開発方法であること。

(C03)［プロトタイピング］プロトタイピングに適した開発方法であること。

(C04)［問題に適した開発方法］その問題に適した開発方法であること。

(C05)［実績のある開発方法］実績のある開発方法であること。

(C06)［マニュアル］その開発方法に対するマニュアルやサービス（サポート体制）が充実していること。

(C07)［習得の容易な開発方法］習得が容易な開発方法であること。

(C08)［ツール］ツールで支援された開発方法であること。

(C09)［形式的な文書化］形式的な文書化が可能な開発方法であること。

(C10)［ＷＳの完備］文書の作成を容易にするワークシート（ＷＳ）が完備していること。

(C11)［チェックリスト］チェックリストが充分に用意された開発方法であること。

(C12)［レビュー］レビューが容易な工程分割になっていること。

(C13)［プロジェクト管理］プロジェクト管理が容易な開発方法であること。

3.1.7.7 ツール・言語選択の適切さの確認

（１）定義

ツールまたは言語選択の選択が適切であったのかどうかを確認することである。

（２）目的と必要性

ツールまたは言語の選択は、知識表現や問題解決・推論方式を決定することとなるので、その選択の良否を評価することは重要である。

（３）説明

エキスパートシステム開発において、実用レベルでのポイントとなるのがツール・言語の選択である。開発全体に与える影響が大きく、各ステップ毎に見直しが要請される。その結果プロトタイプ版と実用版で異なるものを用いる場合も多い。

ツール・言語の選択肢は次のように分類できる。

①ハイブリットツール
②領域特化シェル
③ＡＩ言語
④従来型言語

ツールの評価の要点は以下の通りである。

①ツールの機能／知識表現
②開発者インタフェース
③利用者インタフェース
④システムインタフェース
⑤ツールの性能
⑥ツールの利用目的
⑦ツールが得意とする分野
⑧開発担当者のスキルと開発期間
⑨他システムとのリンク
⑩移植性、保守性
⑪開発環境／デリバリ環境

なお、ツールの詳細評価は（ICOT-JIPDEC AIセンター：知的情報処理システムに関する調査研究報告書、　第２分冊：知識システム構築ツールの評価、平成元年３月）を参照のこと。

（４）点検項目(C)・測定項目(M)

(C01)［要求条件］ツールのサービス、サポート体制、使用契約、価格が要求条件に適合していること。

(C02)［一貫した開発］開発から実運用までの一貫性を考慮して、ツール・言語の選択を検討すること。

(C03)［高性能のツール選択］ツール・言語の選定に当たっては、下記の項目を考慮すること。

- ツールの特徴、開発思想
- ツールの機能／知識表現（知識表現レベル、ワーキングレベルの扱い、推論制御、問題解決レベル、アクセス制御）
- 開発者／利用者ｲﾝﾀﾌｪｰｽ（編集機能・表示機能と操作性、自然言語処理、説明機能、デバッグ機能、コンパイル機能、

知識獲得支援機能、マニュアル）

- システムインタフェース（多言語、ﾃﾞｰﾀﾍﾞｰｽ、ｾﾝｻﾍﾞｰｽ、ﾈｯﾄﾜｰｸ、入出力ﾃﾞﾊﾞｲｽ、ｱﾌﾟﾘｹｰｼｮﾝﾊﾟｯｹｰｼﾞ）
- ツールの性能（実行速度、利用環境、メモリ管理、知識ベースの容量の制約、価格）
- ツールの利用目的（利用レベル、利用者レベル、開発プロセスでの利用性）
- ツールが得意とする分野の特性（製品分類、問題のタイプ、適用例、ツールがサポートできる基本タスク、実績）
- 開発環境／デリバリ環境

(C04)［使用しない言語機能］ツールで提供する機能で使用することがない機能があるのかをチェックすること。
あまりにもオーバースペックの場合は、再検討する必要がある。

(C05)［仕様変更要求］開発担当者が提示したそのツールへの仕様変更要求が頻繁にないこと。
かなりある場合は、再検討する必要がある。

(C06)［システムダウン］開発中に発生したシステムダウンがないこと。
システムダウンが頻繁にある場合は、そのツールの完成度が低いと言える。

(C07)［要求条件］［ツールの制約］ツール基本部の大きさと要求条件との比較をすること（メモリ、ディスク容量、スワップ領域など）。

《その他の評価（経済性、社会性）》

3.1.7.8 システム開発の生産性評価

（１）定義

エキスパートシステムの開発作業の生産性について評価することてある。

（２）目的と必要性

エキスパートシステムの開発には、プロトタイピング手法の利用、ツールの活用などの特徴がある。それらの効果の度合を明確にすることは、その後の開発業務の企画や実施において貴重な情報となる。

（３）説明

エキスパートシステムは、それが用いてる基本的な手法によって、「完全な知識＋演繹推論」（良構造な問題）（以後、タイプＩと呼ふ）と、「不完全な知識＋高次推論」（悪構造な問題）（以後、タイプⅡと呼ぶ）に分けられる。

タイプＩのエキスパートシステムは、通常、ルールベースシステムやフレームベースシステムを採用して構築されるが、従来のソフトウェア開発技術でも十分に構築可能であるので、従来技術に比へ、エキスパートンステム開発技術を採用したことにより、どの位生産性が向上したかの観点で定量的に評価することが適切であると思われる。

タイプⅡのエキスパートシステムは、そもそも悪構造な問題を対象としているので、問題構造や知識の曖昧性を扱う必要があり、従来のソフトウェア開発技術では手に負えなかったり、あるいは、うまく構築することができない。従ってエキスパートンステム開発技術を採用して構築せざるをえず、従来技術と比較しての生産性という観点で評価するのは不適切である。むしろ成長性の観点をも考慮に入れて、長期的な生産性を評価することが必要である。

（４）点検項目(C)・測定項目(M)

(C01)［生産性指標］システムの開発の生産性を測る指標（時間、コスト、開発環境）を前もって設定すること。

(C02)［生産性の目標］目標とする生産性を決定すること。

(C03)［開発・保守の生産性］その生産性目標は妥当であること。

(C04)［開発手法の有効性］ルールベースシステムやフレームベースンステムがソフトウェア開発手法としてどの程度有効であるのかの検討すること。

(C05)［生産性の算出］生産性の算出の際には、原則として、直接の開発従事者だけではなく、開発に関連した要員すへて（特に、専門家、エントユーザ）を考慮にいれること。

(M01)１個の知識の獲得に要した平均時間

(M02)１個の知識の獲得に要した平均コスト

(M03)１個の知識のコーディングに要した平均時間（含：文書化、コーディング、テスト）

(M04)１個の知識のコーディングに要した平均コスト（含：文書化、コーディング、テスト）

(M05)総ルール数／総開発工数（人日）

(M06)総ルール数／総開発費

(M07)総ステップ数／総開発工数（人日）
(M08)総ステップ数／総開発費
(M09)総機能数／総開発工数（人日）
(M10)総機能数／総開発費
(M11)知識ベースの拡張に要したリソース（時間、コスト、工数）

3179 システムのコストパフォーマンス評価

（1）定義

エキスパートシステムに対して投資したトータルコスト（開発コスト、運用コスト、保守コスト）を評価することである。

（2）目的と必要性

企業においては、システムのコストパフォーマンスは重大な関心事である。情報処理システムに関しても、手法や手段によってその差が大きいと想定される場合には、適切な比較評価を行い、その後の計画に充分活用することが必要である。

（3）説明

基本的な評価項目は、以下の通りである。

①エキスパートシステム導入後、経費の節減程度

②作業の質の改善

③生産性の向上

また、利点と弊害という観点で評価する必要がある。

①実用システム導入による利点

詳細は、表31.7-1を参照のこと。

②導入により発生する可能性のある弊害

- 職場モラルの低下、拒否反応
- 定型の結果しか得られない
- 責任の所在が不明確
- 技術のブラックボックス化
- 慣れるのに手間、時間がかかる
- エキスパートシステムによる自動的判断の危険性

（4）点検項目(C)・測定項目(M)

(C01)［経費の節減］エキスパートシステム導入後、以前に比べて経費は節減されたか、あるいは増加したのかを検討すること。

(C02)［作業の質］投資に見合った、作業の質が改善されていること。

(C03)［生産性の向上］投資に見合った、生産性の向上が得られていること。

(C04)［コスト効果］実用システム導入による利点について、表317-1の観点から充分検討を行い、どの程度のコスト効果があったのかを検討すること。

(C05)［コスト換算］人手の場合の生産性とシステムを利用した場合の生産性をコスト換算して比較すること。

(C06)［導入費用］導入費用（マシン、言語等）を積算すること。

(C07)［運用費用］運用費用を積算すること。

3 1 7.10 セキュリティ対策の評価

（１）定義

エキスパートシステムを運用するに際して想定されるセキュリティの侵害に対し、どのような予防措置、対策が講じられているのかを評価することである。

（２）目的と必要性

セキュリティ対策の不足により侵害を受け破壊した場合、システムへのダメージが大きいので、その評価は重要である。

（３）説明

他の情報処理システムと同様、エキスパートシステムにおいても信頼性、安全性の両面からのセキュリティ機能の完備は必要不可欠である。

従来のシステムでは、すでにデータの消失、改ざん、漏洩などによる多くの事故が発生しており、今後ますます重要な問題に発展しつつある。特に、エキスパートシステムには貴重な知識が蓄積されており、改ざんされたり、悪用された場合には従来以上のセキュリティ問題が発生する可能性もある。

セキュリティ対策上留意すべき点は以下の通りである。

①リスク分析の必要性

特に、エキスパートシステムが社会システムやその他のシリアスな対象に用いられた場合は、まず起こり得るリスクを事前に充分検討しなければならない。例えば、解答が誤った場合、利用者の財産、生命に危具を与え、取り返しのつかない事態を招く恐れもある。

②セキュリティ対策には、技術的対策、運用上の対策、設備上の対策などがある。全ての角度から充分な対策を講ずるべきである。

③システム自体の実行権および知識ベースへのアクセス権を厳格に設定し、充分なアクセスコントロールを実施すること。

④不正アクセスや改ざんが行われたか否かを検査する手段を設けること。

⑤ネットワークのデータ伝送路上での充分なセキュリティ対策を講じること。

⑥知識、ノウハウに対する知的所有権、侵害を受けた場合の法的責任などについても事前に明確にしておくことが必要である。

（４）点検項目(C)・測定項目(M)

(C01)［充分な対策］セキュリティ対策には、技術的対策、運用上の対策、設備上の対策などがあるが、全ての角度から充分な対策を講じること。

(C02)［リスクの事前検討］特に、社会システムやその他のシリアスな対象に用いられるエキスパートシステムの場合は、まず起こり得るリスクを事前に充分に検討すること。

例えば、解答が誤った場合、利用者の財産、生命に危具を与え、取り返しのつかない事態を招く恐れもある。

(C03)［知的所有権］［法的権利］知識、ノウハウに対する知的所有権、侵害を受けた場合の法的権利などについても事前に明確にすること。

(C04)［利用者の範囲］利用者（社内／社外／不特定多数）の範囲の検討を行うこと。

(C05)［情報の開示レベル］情報の開示レベルを検討すること。

(C06)［内容の改ざん］知識ベースは理解性が高いために、逆に内容が漏洩したり改ざんされたりする危険性が高いことに注意すること。

(C07)［設計ノウハウの隠蔽］設計ノウハウの隠蔽（ルールコンパイル等）対策をすること。

(C08)［アクセスコントロール］システム自体の実行権および知識ベースへのアクセス権を厳格に設定し、充分なアクセスコントロールを実施すること。

(C09)［不正アクセス］不正アクセスや改ざんが行われたか否かを検査する手段を設けること。

(C10)［データ伝送路］ネットワークのデータ伝送路上での充分なセキュリティ対策を講じること。

《参考文献》

[Kahn 1988] Gary Kahn, Neil Jacobstein: Managing Expert Systems Projects, AAAI88, TUTORIAL, MP3, 1988

[Walters 1988] John Walters,Norman R Nielsen: Crafting Knowledge-Based Systems, JOHN WILEY&SONS, 1988

[Warfield 1974] Warfield,J N Structuring Complex Systems, Battelle Monograph 4, (April 1974)

[AIｾﾝﾀｰ 1986] ICOT-JIPDEC AIセンター: 昭和61年度知的情報処理システムに関する調査研究報告書、知識システム開発方法論、昭和62年9月

[AIｾﾝﾀｰ 1987] ICOT-JIPDEC AIセンター: 昭和62年度人工知能技術動向等に関する調査研究報告書、付属資料（エキスパートシステム事例）、昭和63年3月

[AIｾﾝﾀｰ 1989] ICOT-JIPDEC AIセンター: 知的情報処理システムに関する調査研究報告書、平成元年3月
第１分冊　計画・設計型知識システムの構築方法論
第２分冊　知識システム構築ツールの評価
第３分冊　知識獲得支援
第４分冊　知識システム技術者育成ガイトライン
第５分冊　知識システムの事例

[秋田 1987] 秋田興一郎: エキスパートシステム導入実戦ガイト、電気書院、昭和63年10月

[阿部 1987] 阿部憲広、滝寛和: エキスパートシステム入門、共立出版、1987年10月

[上野 1987] 上野晴樹、石塚満編: 知識の表現と利用、昭和63年2月、ｵｰﾑ社

[小林 1987] 小林重信: 知識工学、昭晃堂、昭和62年10月

[金剛寺 1990] 金剛寺、鍛治、寺野: 人工知能システムが実用的であることについて、知識工学と人工知能 71-1-2、第７１回人工知能研究会、情報処理学会、７月、1990

[JIPDEC 1989] 日本情報処理開発協会中央情報教育研究所（ＣＡＩＴ）編: 高度情報処理技術者育成指針－ナレッジエンジニア編－、平成元年3月

[情報処理 1987] 特集エキスパートシステム、情報処理、Vol 28, No 2, 1987年2月

[人工知能学会 1988] 昭和63年度人工知能学会全国大会（第２回）論文集、7 21-7 23、人工知能学会

[末田 1989] 末田直道、他: エキスパートシステム構築技法ＴＵＰＰＳ－ＥＳ、東芝レビュー、Vol 44, No 10, pp 814-817, 1989

[中尾 1980] 中尾、ほか: システム計画のためのシステム分析手法“ＰＰＤＳ”の開発、日立評論、No 62、pp 21-28、12月、1980

[日立 1986] 日立製作所．システム開発標準　エキスパートシステム構築標準手順ＥＳＧＵＩＤＥ（第１版）、昭和62年10月

[富士通 1986] 富士通: Ｃ－ＮＡＰ（手順編）、ＶＴＲ補助テキスト、UAS23V1N*1、1986

[富士通 1988a] 富士通: ＥＳ／ＳＤＥＭエキスパートシステムの構築技法、昭和63年

[富士通 1988b] 富士通：エキスパートシステムの構築技法、テキスト、昭和63年7月.

[ﾍｲｽﾞﾛｽ 1985] ﾌﾚﾃﾞﾘｯｸﾍｲｽﾞｰﾛｽ、ﾄﾞﾅﾙﾄﾞ A ｳｫｰﾀｰﾏﾝ、ﾀﾞｸﾞﾗｽ B ﾚﾅｰﾄ：エキスパートシステム、ＡＩＵＥＯ訳、産業図書、昭和60年8月

[松村 1983] 松村、ほか：要求分析技法、東芝レビュー、Vol 38、No 11、1983年11月

[松本 1989] 松本吉弘：情報処理ハントブック、Ⅱ編ソフトウェア工学、２章仕様化、pp 967-977、1989年5月、オーム社

[山口 1988] 山口高平：分化し始めたＡＩツール、TURING MACHINE、Vol 1、No 1、1988年12月

３ ２ ライフサイクル・評価者別評価項目

本節では、エキスパートシステムをライフサイクルに則って評価する際のポイントと、実際の評価をする人の役割や関心事に則っての評価のポイントについて述べる。

表3.2-1に、エキスパートシステムのライフサイクル・評価者別の重点評価項目を示す。すべての評価項目に対してすべてのフェーズて評価すへきではあるが、表中の○印は特にこのフェーズで重点的に評価すべきであると考えられる評価項目に対して付けられている。なお、○印の付いていない評価項目に関しては、このフェーズで評価しなくてもいいといっているのではなく、そのエキスパートシステムの重大度（リスク、安全性の面）・特質や開発・評価コストなどを勘案して、評価者各自がどの評価項目を選択し、実施すへきかを事前に決めておく必要がある。しかしながら、エキスパートシステムのライフサイクルの特質上、知識獲得はすへてのフェーズで行われ、知識ヘースが更新・洗練化されており、常にエキスパートシステムの品質を評価・確認する必要があるので、すへてのフェーズで当該評価項目に対して展開された点検項目および測定項目（３ １参照）に関しては、原則として、すへて点検・測定することが望ましい。各評価者は、評価作業でのその人の役割分担や責任分担に応して評価項目を適宜分割・分担してもよいが、品質確保の点からは、その品質が多面的に評価されるので、むしろ重複は望ましいと考えられる。

３．２．１　ライフサイクル別評価のポイント

ライフサイクル別の評価ては、当該フェースの目的・目標が達成されたかに重点を置いて実施する必要がある。そのためには、各フェーズの開始時点て、そのフェーズの目的・目標を明確に設定し、文書化（各フェースの要求仕様なと）しておく必要がある。またプロトタイピンクフェーズでも、プロトタイプ（プロクラムコード）たけでなく、レビューやインスペクションに耐えうるトキュメントも作成する必要がある。

なお、各フェーズの説明は、２５２項を参照のこと。

３２１１知識ベースの妥当性評価

知識ヘースの内容は、フェーズの進展に伴って充実し、正当性、完備性、一貫性がそれぞれ向上していく。一般的には、プロトタイピングフェースでは正当性、実用システム開発および運用テストのフェースでは一貫性、そして運用・洗練化フェーズでは完備性の評価に重点を置いて行われると言えよう。しかし、知識が知識ヘースに追加された際には、その都度、その知識の正当性を評価するのはもちろんのこと、知識ヘースの一貫性についても評価する必要がある。知識表現に関しては、一度決定すると変更が困難であるので、プロトタイピングフェースで重点的に評価を行う必要がある。

（１）知識ベースの正当性評価

知識ベースの正当性評価はすべてのフェーズでの最重点評価項目てある。プロトタイピングフェーズでは一般化された知識が、そして運用・洗練化フェースでは特殊化された知識が主たる対象となろう。またプロトタイピング／実用システ

表32-1 ライフサイクル別評価者別による重点評価項目（つづく）

分類	評価項目 ＼ 評価者 ＼ 開発フェーズ	調査・計画			プロトタイピング							実用システム開発					運用テスト						運用・洗練化		
					実験型		発展型																		
		マネージャ	ユーザ	専門家	開発担当者	専門家	開発担当者	専門家	第3者専門家	ユーザ	マネージャ	開発担当者	専門家	第3者専門家	ユーザ	マネージャ	開発担当者	ユーザ	運用担当者	専門家	第3者専門家	マネージャ	ユーザ	運用担当者	マネージャ
知識ベースの妥当性評価	(A)知識ベースの正当性評価 [P]						○	○	○			○	○	○			○			○	○			○	
	(B)知識ベースの完備性評価 [P]											○		○			○				○			○	
	(C)知識ベースの一貫性評価 [P]						○					○					○							○	
	(D)経験的知識を正当化する方法とその確認 [E]						○	○	○																
	(E)知識表現の選択と適切さの確認 [D]				○		○					△													
	(F)知識獲得方法の選択と適切さの確認 [D]				○	○	○	○	○																
問題解決・推論方式の妥当性評価	(A)推論処理の妥当性評価 [P]				○		○	○	○			△													
	(B)推論時間の妥当性評価 [P]						○			○		○			○		○	○	○				○	○	
	(C)推論手法の選択と適切さの確認 [D]				○		○					△													
	(D)不確実情報に対する処理技法の適切さの確認 [D]				○		○	○	○			△													
	(E)学習機能の適切さの確認 [D]				○		○					△													
結果の妥当性評価	(A)正解の評価 [P]				○	○	○	○	○			○	○	○			○	○	○	○	○		○	○	
	(B)解の網羅性評価 [P]											○	○	○			○	○	○	○	○		○	○	
	(C)解の最適性評価 [P]											○	○	○			○	○	○	○	○		○	○	
	(D)解の評価方法の適切さの確認 [E]				○		○	○	○																
要求仕様に対する達成度評価	(A)要求仕様内容に対する評価 [P]	○	○		○		○					○			○	○		○					○	○	○
	(B)要求仕様に対する達成度評価 [P]				○		○			○	○	○				○		○	○			○	○	○	○
	(C)要求仕様設定手段の確認 [D]											○													

P プロダクトに対する評価　E 評価方法に対する評価　D 開発/保守方法 技法・ノールに関する評価　O その他　△ 再開発の場合

分類	評価項目 ＼ 評価者 ＼ 開発フェーズ	調査・計画			プロトタイピング 実験型		プロトタイピング 発展型					実用システム開発					運用テスト						運用・洗練化		
		マネージャ	ユーザ	専門家	開発担当者	専門家	開発担当者	専門家	第3者専門家	ユーザ	マネージャ	開発担当者	専門家	第3者専門家	ユーザ	マネージャ	開発担当者	ユーザ	運用担当者	専門家	第3者専門家	マネージャ	ユーザ	運用担当者	マネージャ
知識ベースの保守性評価	(A)変更箇所発見の容易性評価 [P]											○													
	(B)知識ベースの理解性評価 [P]											○													
	(C)変更箇所の局所性評価 [P]											○													
	(D)知識ベースの統一性評価 [P]											○													
	(E)品質安定性の確認方法の確認 [D]											○							○						
	(F)保守マニュアルの整備状況の確認 [D]											○							○				○	○	
	(G)保守用ノールの整備状況の確認 [D]											○							○				○	○	
HIF妥当性の評価	(A)利用時のインタフェース評価 [P]						○					○						○	○				○	○	
	(B)保守時のインタフェース評価 [P]											○						○	○				○	○	
システムの総合評価	(A)テーマ選定の妥当性評価 [P]	○		○			○	○	○		○														
	(B)利用者の満足度評価 [P]									○	○							○	○			○	○		○
	(C)統合性の評価 [P]	○										○					○	○	○						
	(D)成長可能性の評価 [P]						○	○	○		○	○	○	○		○				○	○	○			○
	(E)開発／保守用ドキュメントの評価 [P]				○		○					○				○			○					○	
	(F)開発方法の適切さの確認 [D]	○			○		○				○	△				△									
	(G)ノール 言語選択の適切さの確認 [D]	○			○		○				○	△													
	(H)システム開発の生産性評価 [O]	○					○				○	○				○									
	(I)システムのコストパフォーマンス評価 [O]	○															○	○	○			○	○	○	○
	(J)セキュリティ対策の評価 [O]	○										○					○		○			○			

ム開発／運用テストのフェーズでは、知識の正当性に関しては、フェーズの終わりでバッチ的に評価されるが、運用・洗練化フェーズては、知識か追加される毎に、直ちにその知識の正当性評価が行われるのが理想的てある。

（２）知識ベースの完備性評価

知識ベースの完備性は、一般に、フェーズの進展に伴って向上する。完備性の評価に当たっては、当該フェースの完備性目標（例えば何％）に基ついて、その目標が達成されたかを評価する必要がある。ただし、一般に、ある程度知識が完備してくると、少々完備率が向上しても、それに見合って問題解決能力は向上しない。

（３）知識ヘースの一貫性評価

知識ベースの一貫性は、一般に、フェーズの進展に伴って劣化する傾向にある。しかし、知識ヘースの一貫性が劣化したからといって、即、問題解決能力が低下するというわけではないので、一貫性の評価は後回し、あるいは無視される可能性があるが、知識ベースの質を保つ上でも、各フェース毎に必ず実施する必要がある。少なくとも、当該フェーズの誤りを当該フェースて検出して修正すれば、後々のフェーズで検出して修正するのに比へ、少ない手間やコストて済む。また一貫性は知識ヘースの保守性に深い関係があり、一貫性の低下の問題は運用・洗練化フェーズで顕在化し、重大な問題となる。

（４）経験的知識を正当化する方法とその確認

知識ベースに格納されている知識の中核は経験的知識であり、経験的知識を正当化する方法を開発フェーズの初期の段階で検討し、その有効性を確認または評価する必要がある。この評価項目に関しては、プロトタイピンクフェースて１回だけ確認または評価すれば、原則として、以後のフェーズては確認または評価する必要がない。

（５）知識表現の選択と適切さの確認

知識表現の選択はエキスパートシステムの問題解決能力や作成しやすさを決定付けるクリティカル要因であるので、すへてのフェーズでその適切さを確認または評価する必要がある。しかし、知識表現の変更には膨大な作業を伴うので、プロトタイピングフェーズで十分な確認または評価を行い、てきるたけ早いフェーズで確固たる根拠に基づいて選択すべき知識表現を決定することが要締てある。

（６）知識獲得方法の選択と適切さの確認

知識獲得はエキスパートシステム開発での中核的作業である。知識獲得方法の良否が、獲得される知識の量と質を決定付けるので、プロトタイピンクフェースでその適切さに関して十分な確認または評価を行う必要がある。

３２１２ 問題解決・推論方式の妥当性評価

問題解決・推論方式はアーキテクチャであり、知識ヘースとは異なり、原則としてフェーズの進展に伴って量が増えたり、変化したりするものではないが、知識ヘースの量の増加に伴って、方式自体の妥当性がかなり影響を受けるので、すへてのフェーズで評価する必要がある。

（１）推論処理の妥当性評価

初期のフェーズでは知識ヘースの量も少量であり、妥当てある推論処理も、フェーズが進むにつれて知識ヘースが大規模化して妥当でなくなることもあるので、各フェーズでは、知識ヘースの量と推論処理との関連を意識して評価することが

肝心である。
（２）推論時間の妥当性評価
推論時間の妥当性に関しては、各フェースで設定された目標の推論時間が達成されたかの観点で評価する。運用テストフェーズや運用・洗練化フェーズでは、エキスパートシステムと他システムとが統合されるので、エキスパートシステムだけの推論時間ではなく、トータルシステムでの応答時間をも評価する必要がある。
（３）推論手法の選択と適切さの確認
推論手法の選択はエキスパートシステムの問題解決能力や推論過程の理解容易性を決定付けるクリティカル要因であるので、すべてのフェーズでその適切さを確認または評価する必要がある。しかし、推論手法の変更は知識ベースの改造を伴う膨大な作業が必要なので、プロトタイピングフェースで十分な確認または評価の上、確固たる根拠に基づいて選択すべき推論手法を決定することが要締である。
（４）不確実情報に対する処理技法の適切さの確認
不確実情報に対する処理技法の選択は、特にプロトタイピングフェース全体にわたって試行錯誤的に決定されるので、その都度評価する必要がある。また、運用・洗練化フェーズでは、不確実情報が、エキスパートシステムの問題解決能力に影響を与える可能性が高いので、その観点で確認または評価する必要がある。
（５）学習機能の適切さの確認
学習機能はエキスパートシステムの処理能力を向上させる有効な手段であるので、プロトタイピングフェーズでその選択された学習機能の適切さを確認または評価する必要がある。

3 2.1 3 結果の妥当性評価

エキスパートシステムの能力、すなわち、結果の妥当性が最も主要な評価項目であるので、すべてのフェーズにわたって十分に、厳密に評価が行われなければならない。
（１）正解の評価
正解の評価はすべてのフェーズで実施されなければならない。特に知識ベースが更新された場合には、新規のテスト問題だけでなく、以前に行ったテスト問題すべてが正しく解けるかについてもテストし、再評価する必要がある。
（２）解の網羅性評価
解の網羅性に関しては、プロトタイピングフェーズでは網羅性は設計目標でないことも多いので、フェーズの進展に伴って網羅性がどの位向上、あるいは成長したかの観点で評価する必要がある。実運用のフェーズでは、最終目標とした網羅度が達成されたかの観点で評価することになる。
（３）解の最適性評価
解の最適性に関しては、プロトタイピングフェーズでは最適性は設計目標でないことも多いので、フェーズの進展に伴って最適性がどの位向上したかの観点で評価する必要がある。実運用のフェーズでは、最終目標とした最適度が達成されたかの観点で評価することになる。
（４）解の評価方法の適切さの確認
エキスパートシステムは悪定義問題や悪構造問題を対象としていることもあり、

正解あるいは最適解が分からないことも多いので、解の評価方法が誤っていると、エキスパートシステムの問題解決能力に関して誤判定をしてしまう可能性がある。それ故に解の評価方法に関しては、開発フェーズの初期の段階て十分に検討し、その適切さを確認または評価する必要がある。

3.2.1 4 要求仕様に対する達成度評価

エキスパートシステムといえども要求仕様を実現するために開発されるので、すへてのフェーズにわたって、そのフェーズで設定された要求仕様が達成されたか否かの観点で評価する必要がある。

（１）要求仕様内容に対する評価

プロトタイピングフェーズでは、そのプロトタイプ開発の目的にかなった要求仕様内容（例えば、実現可能性検証、技術検証、設計検証、能力検証など）になっているかの観点で評価する。そして、実用システム開発フェースては、ンステム化の目的・目標やユーザニーズに合っているかの観点て要求仕様内容を評価することになる。各フェーズでの要求仕様項目については、２６２項を参昭のこと。

（２）要求仕様に対する達成度評価

各フェーズ毎の要求仕様書が存在することが前提であるので、それぞれのフェーズでその達成度を評価する必要がある。

（３）要求仕様設定手段の確認

要求仕様は開発・評価に係わるあらゆる作業の拠り所であるので、そのためには要求仕様設定手段が適切で、有効である必要がある。要求仕様が最も重要な段階は実用システム開発フェーズであるので、そのフェースで確認または評価を行う必要がある。

３２１５ 知識ベースの保守性評価

知識ヘースはフェーズの進展に伴い成長するので、その保守性の評価はエキスパートシステムの問題解決能力評価とは別の面で、即ち、エンジニアリンクの面で特に重要である。プロトタイピングフェーズでは、知識ヘースへの知識の追加・更新（即ち、保守）は、開発担当者が行うが、運用・洗練化フェースでは、運用担当者やユーザが行うので、実用システム開発フェーズて重点的に保守性を評価することになる。

実用ンステムがプロトタイプの改善によって作成される場合には、一般にプロトタイプ開発では保守性について考慮していないので、保守性の面ての改善が充分に行われたかの点について重点的に評価する必要かある。

（１）変更箇所発見の容易性評価

知識ヘースの保守は継続的に行われるため、その変更箇所の発見が容易てあることが必要となる。これは特に第３者保守の対象となる実用ンステム開発フェーズの終了時点での評価が重要である。

（２）知識ベースの理解性評価

知識ベースを正しく保守するためには、その理解性が重要てある。これも第３者保守の対象となる実用ンステム開発フェーズの終了時点ての評価が重要てある。

（３）変更箇所の局所性評価

保守の容易性のためには、変更箇所が局所的であることが必要となる。これに

関しても、第３者保守の対象となる実用ンステム開発フェースての評価が重要である。
（４）知識ベースの統一性評価
知識ヘースの品質要件の１つに統一性がある。これに関しても、実用ンステム開発フェーズの評価が中心となる。
（５）品質安定性の確認方法の確認
知識ヘースは常に更新されるので、その都度品質の安定性を確認する必要がある。そのための確認方法が用意され、有効であるかの面について、実用ンステム開発フェーズと運用テストフェーズにわたって確認または評価する必要がある。
（６）保守マニュアルの整備状況の確認
保守の際に必要となる保守マニュアルに関しては、まず実用ンステム開発フェーズで、その内容について確認または評価する必要がある。しかし、運用テストや運用段階での改善が必要となることも多いため、これらのフェースての評価も必要である。
（７）保守用ツールの整備状況の確認
保守用ツールの種類や機能に関しては、実用システム開発フェースて確認・評価する必要がある。運用テストや運用・洗練化フェースてさらに改善する必要かるので、それらのフェーズでの評価も必要である。

３２１６ヒューマンインタフェースの妥当性評価

ヒューマンインタフェースは利用者や保守者の作業効率に直接影響を与えるのみでなく、その良否はそのンステムが利用されるか否かを左右する場合もあるため、特に実用ンステムては十分評価する必要がある。
（１）利用時のインタフェース評価
エキスパートンステムが実際に使用されるかのクリティカルポイントの１つが利用時のインタフェースであるので、プロトタイピングフェーズても評価する必要がある。
（２）保守時のインタフェース評価
知識ヘースは常に更新されるので、保守時のインタフェースの良否が保守時の作業効率や品質の安定性に直接影響を与える。特に第三者保守が発生する実用ンステム開発フェーズ以降ての評価が重要である。

３２１７ンステムの総合評価

システムの総合評価は、評価項目に応して、多様なフェースての評価が必要となる。
（１）テーマ選定の妥当性評価
テーマ選定の妥当性に関しては、ライフサイクルの初期の時点て充分に評価する必要がある。この評価結果はエキスパートンステムの開発が成功するか否か、あるいはエキスパートシステム開発をやるへきかを決定付けるので、重点評価項目として行う必要がある。
（２）利用者の満足度評価
利用者の満足度評価は多くのフェースで継続的に行い、その評価結果に基づいて、エキスパートンステム全体を改善する必要がある。

（３）統合性の評価

エキスパートシステムの実用性を高めるポイントは他システムとの統合であるので、実用システム開発および運用テストフェースで重点的に評価する必要がある。

（４）成長可能性の評価

エキスパートシステムでは、その成長可能性を見極めることが非常に大切であり、すべてのフェースで継続的に評価を行う必要がある。そのフェースで達成された問題解決能力を基に、後続フェースでの問題解決能力の成長可能性を総合的に検討・評価するが、初期のフェーズではラフに、そして後期のフェーズでは厳密に行う必要がある。

（５）開発／保守用トキュメントの評価

開発／保守用トキュメントは、すべてのフェーズで十分に評価する必要がある。

（６）開発方法の適切さの確認

開発方法は開発作業の効率や質に直接影響を与えるクリティカル要因であるので、ライフサイクルの初期の段階で、その適切さについて十分に確認または評価する必要がある。

（７）ツール・言語選択の適切さの確認

ツール・言語は、エキスパートシステムでもっとも重要である知識表現や推論方式の選択肢を制約あるいは決定付けることになるので、その適切さに関しては、ライフサイクルの初期の段階で重点的に確認または評価する必要がある。

（８）システム開発の生産性評価

システム開発の生産性向上は、ソフトウェア開発での中心課題であるので、調査・計画フェーズでの評価が重要であるとともに、以後もそれぞれのフェースに則した評価と、その評価結果に基づいた改善が必要である。

（９）システムのコストパフォーマンス評価

本ガイトラインが対象とする実運用型エキスパートシステムのシステム化の最大の目的・目標はコストパフォーマンスの向上であるので、調査・計画フェースとシステムがある程度完成した運用テストおよび運用・洗練化フェースで評価する必要がある。

（10）セキュリティ対策の評価

エキスパートシステムには、その組織での高度な技術的ノウハウが大量に蓄積されているので、セキュリティ対策に関しては、すべてのフェースで継続的に評価する必要がある。そして、その評価結果に基づいて、セキュリティ対策を改善する必要がある。

３２２　評価者別評価のポイント

各評価者は評価での責任分担範囲、役割、各自の関心事などに則って評価すべき対象項目・対象範囲を事前に決め、各評価者は原則として独立に、それぞれの独自の観点で評価を行う必要がある。

ここでは、評価者として下記に示す人達を想定する。

- 開発担当者　：開発グループの中でも、主として段階毎のデザインレビューや、ウォークスルーを行う人達である。
- 専門家　　：知識提供者である。

- 第３者専門家：当該分野の専門家で、知識提供者とは異なる人てある。
- ユーザ　　：専門家支援ンステムでは、ユーザは専門家て、専門家の代行ンステムでは、ユーザは非専門家てある。
- 運用担当者　：実用システムの管理・運用を行い、ンステムの変更の責任を担う人達てある。
- マネージャ　：次のフェーズへ進む決断を行う人である。
- 企画グループ：企画グループは、ンステム開発に先立ち、ンステムの調査・計画を行うグループであり、次のメンバーから構成されている。
  - 利用部門のマネージャ
  - 開発部門のマネージャ
  - 運用部門のマネージャ
  - エントユーザ
  - ドメインの専門家

3 2.2.1 知識ベースの妥当性評価

知識ベースの妥当性の評価は、主として、エキスパートシステム化の対象分野に精通した専門家である、知識を提供した専門家とそれ以外の第３者専門家とが実施する。知識のコート化（コンピュータ化）に関する評価ては、開発担当者が主体となって実施する。

（１）知識ベースの正当性評価

知識ベースの正当性評価の主実施者は、エキスパートシステムては、あくまても、知識を提供した専門家自身であり、それを補完する立場で第３者専門家が評価を行う。知識ベースの正当性評価に関しては、開発担当者は助手という立場て評価に協力する。知識ベースの正当性評価では、専門家（人間）によるレビューやインスペクションが主体ではあるが、実験や深い知識からの知識コンパイルなどによる確認・実証も重要である。

［評価者の特徴的な点検項目］

＊開発担当者　：(*C1)獲得した知識に対してオリジナルの知識源に遡って、その正しさを点検すること。

＊専門家　　：(*C2)自分が提供した知識が抽象化された観点で見て、トキュメント上に正しく再現されているかを点検すること。

＊第３者専門家：(*C3)専門家から獲得された経験的知識が第３者から見て、妥当てあるかを点検すること。

＊運用担当者　：(*C4)トキュメント上の知識と知識ベース上の知識とが同してあるかを点検すること。

（２）知識ベースの完備性評価

知識ベースの完備性に関する論理的な誤りはツール（コンピュータ）により自動的に検出可能であるので、開発担当者でも評価を行うことができる。しかし、知識がどの程度まで完備していれば、どの程度まで・どの範囲までの問題が解けるかに関しては、第３者専門家に評価を行ってもらう必要がある。

［各評価者の特徴的な点検項目］

＊開発担当者　：(*C1)静的分析ツールを使って知識ベースの論理的な完備性を点検すること。

＊第３者専門家：(*C2)要求仕様に規定された範囲の問題を解くのに必要な知識がすへて完備しているかを点検すること。

＊運用担当者　：(*C3)すべての知識が正しくトキュメント化されているかを点検すること。

（３）知識ベースの一貫性評価

知識ベースの一貫性に関する論理的な誤りはノール（コンピュータ）により自動的に検出可能であるので、開発担当者でも評価を行うことがてきる。一貫性は主としてコート化の際の問題てあるので、開発担当者が主体となって、インスペクションなどの手段を用いて評価が行われる。

［各評価者の特徴的な点検項目］

＊開発担当者　：(*C1)静的分析ノールを使って知識ベースの論理的な一貫性を点検すること。

＊運用担当者　：(*C2)知識ベースおよびそのトキュメントがコーディンク標準およびトキュメント標準に準拠しているかを点検すること。

（４）経験的知識を正当化する方法とその確認

経験的知識を正当化する方法に関しては、知識ベースに格納されている経験的知識は対象領域固有のものであるので、正当化する方法も領域固有てあると考えられる。従って、その領域の専門家が主体となって確認または評価する必要がある。

［各評価者の特徴的な点検項目］

＊開発担当者　：(*C1)正当化する方法に関して網羅的に検討したかを点検すること。

＊専門家　　：(*C2)その正当化する方法でよいかを点検すること。

＊第３者専門家：(*C3)その領域でよく使われる正当化する方法が使われているかの観点で点検すること。

（５）知識表現の選択と適切さの確認

知識表現の選択はコンピュータ化固有の問題であるので、開発担当者が実施し、その適切さを確認または評価する必要がある。

［各評価者の特徴的な点検項目］

＊開発担当者　：（３　１節て指摘された該当点検項目すへて）

（６）知識獲得方法の選択と適切さの確認

知識獲得方法の選択に関しては、知識獲得がスムースに、効率的に行われる必要があるので、知識提供側である専門家と知識獲得側てある開発担当者とが協力して、その適切さを確認または評価する必要がある。

［各評価者の特徴的な点検項目］

＊開発担当者　：(*C1)知識獲得が能率的に行われたかを点検すること。

＊専門家　　：(*C2)ＫＥ（Knowledge Engineer）のインタヒューの仕方が適切であったかを点検すること。

＊第３者専門家：(*C3)知識を提供した専門家が多面的に見て妥当てあるかを点検すること。

３２２２問題解決・推論方式の妥当性評価

問題解決・推論方式の妥当性評価に関しては、問題解決・推論方式か凡用的てる部分とその領域に固有の部分とで成り立つているので、主として専門家と開発担当者が協力して評価を行うことになる。

（１）推論処理の妥当性評価

推論処理の妥当性に関しては、その領域での問題固有の推論処理の部分は専門家が、そしてジェネリックタスク的な汎用性の高い推論処理の部分は開発担当者がそれぞれ分担して評価する必要がある。

［各評価者の特徴的な点検項目］

＊開発担当者　・(*C1)専門家から獲得した問題解決・推論のやり方をノールが提供している問題解決・推論方式にうまく展開てき、かつプログラム上に正しく記述・コーディングされているかを点検すること。

＊専門家　(*C2)自分が提供した問題解決・推論のやり方か抽象化された観点で見て、トキュメント上に正しく再現されているかを点検すること。

＊第３者専門家：(*C3)専門家が提供した問題解決・推論のやり方が第３者から見て妥当であるかを点検すること。

（２）推論時間の妥当性評価

推論時間に関しては、直接実感され、計測できるので、プロトタイピンクフェーズでは推論処理の内部に立ち入って、なぜ推論時間がかかりすぎるかの観点て、専門家と開発担当者が協力して、綿密に計測・評価する。実用システム開発フェーズ以降では外部から推論処理の時間を測定して、ユーザや運用担当者をも交えて、推論時間の要件を基にして評価する必要がある。

［各評価者の特徴的な点検項目］

＊開発担当者　：(*C1)どの場合に長時間の推論を要するかを点検すること。

＊ユーザ　：(*C2)その推論時間て自分の仕事に差し支えないかを点検すること。

＊運用担当者　：(*C3)その推論時間が要求仕様を満たしているかを点検すること。

（３）推論手法の選択と適切さの確認

推論手法の選択はコンピュータ化固有の問題であるので、その適切さを開発担当者自身で検討し、確認または評価する必要がある。

［各評価者の特徴的な点検項目］

＊開発担当者　：（３　１節で指摘された該当点検項目すへて）

（４）不確実情報に対する処理技法の適切さの確認

不確実情報に対する処理技法に関しては、不確実情報の処理がその領域固有てある場合も多いので、専門家と開発担当者が協力して、その適切さを確認または評価する必要がある。

［各評価者の特徴的な点検項目］

＊開発担当者　：(*C1)専門家から獲得した不確実情報に対する処理のやり方をノールが提供している不確実情報に対する処理技法てうまく実現でき、かつプログラム上に正しくコーディングされているかを点検すること。

＊専門家　：(*C2)自分が提供した不確実情報に対する処理のやり方が抽象化された観点で見て、トキュメント上に正しく再現されているかを点検すること。

＊第３者専門家　(*C3)専門家が提供した不確実情報に対する処理のやり方が第３者から見て、その領域ての慣習に合っているかを点検すること。

（５）学習機能の適切さの確認

学習機能はＡＩ固有の問題てあるので、開発担当者のみて、その適切さを確認

または評価することになる。
［各評価者の特徴的な点検項目］
＊開発担当者　：（３　１節て指摘された該当点検項目すへて）

３２.２３結果の妥当性評価

結果の妥当性は、エキスパートシステム開発・利用に関係するすへての人の関心事であるので、原則として、全員で確認または評価する必要がある。
（１）正解の評価
エキスパートシステムが導出した解が正解であるか否かはすへての人の関心事である。テスト問題に関しては全員が協力してバラエティに富んた問題を選定・用意し、そのテスト問題の想定される正解に関しては専門家あるいは第３者専門家が合意して用意する。そしてエキスパートシステムの解が合っているか、誤っているかの評価は全員で総合的に行う必要がある。もし、正解が事前に分からない場合には、専門家あるいは第３者専門家が推論過程をトレースして確認し、当該推論過程そのものが正しければ、その解（結果）も正しいとして評価することになろう。
［各評価者の特徴的な点検項目］
＊開発担当者　：(*C1)テストデータの量と質が統計的に見て妥当てあるか、その正解率が正しく計算されているかを点検すること。
＊専門家　　：(*C2)エキスパートシステムの結果がどの程度正しいかを個別に点検すること。
＊第３者専門家：(*C3)専門家が提供した正解がどの程度正しいかを個別に点検すること。
＊ユーザ　　：(*C4)テストデータが自分の仕事をどの程度反映しているか、その正解率でエキスパートシステムを仕事に使えるかの観点て点検すること。
＊運用担当者　：(*C5)テストデータが運用プロフィールをとの程度反映しているか、その正解率でエキスパートンステムを実用ンステムとして提供てきるかの観点で点検すること。
（２）解の網羅性評価
解の網羅性に関しては、実用システム開発フェーズ以降で、各フェースでの網羅性目標に基づいて、専門家が主体となって評価する必要がある。
［各評価者の特徴的な点検項目］
＊開発担当者　：(*C1)想定した網羅率が達成されたかの観点で点検すること。
＊専門家　　：(*C2)エキスパートシステムが自分とほぼ同し網羅率を達成しているかの観点で点検すること。
＊第３者専門家：(*C3)その領域での網羅率が客観的にとの位であるかの観点で、エキスパートシステムの網羅率を点検すること。
＊ユーザ　　：(*C4)その網羅率でエキスパートシステムを仕事に使えるかの観点で点検すること。
＊運用担当者　：(*C5)その網羅率が要求仕様を満たしているかの観点て点検すること。
（３）解の最適性評価
解の最適性に関しては、実用システム開発フェーズ以降て、各フェースての最

適度目標に基づいて、専門家が主体となって評価する必要がある。
［各評価者の特徴的な点検項目］
＊開発担当者　：(＊C1)想定した最適度が達成されたかの観点て点検すること。
＊専門家　　：(＊C2)エキスパートシステムが自分とほぼ同し最適度を達成しているかの観点で点検すること。
＊第３者専門家：(＊C3)その領域ての最適度が客観的にとの位てあるかの観点て、エキスパートシステムの最適度を点検すること。
＊ユーザ　　：(＊C4)その最適度てエキスパートシステムを仕事に使えるかの観点で点検すること。
＊運用担当者　：(＊C5)その最適度が要求仕様を満たしていかの観点て点検すること。
（４）解の評価方法の適切さの確認
解の評価方法に関しては、専門家、第３者専門家、そして開発担当者の３者て検討・合意して、その３者てその適切さを確認または評価することになる。
［各評価者の特徴的な点検項目］
＊開発担当者　：(＊C1)評価方法に関して網羅的に検討したかを点検すること。
＊専門家　　：(＊C2)その評価方法てよいかを点検すること。
＊第３者専門家：(＊C3)その領域てよく使われる評価方法が使われているかの観点て点検すること。

3 2 2.4 要求仕様に対する達成度評価

要求仕様はすへての評価の際の拠り所となるものであり、そしてエキスパートシステム開発・利用に関係するすへての人の関心事てあるので、原則として、全員で精査し、確認または評価する必要がある。
（１）要求仕様内容に対する評価
プロトタイピングフェースての要求仕様は、プロトタイピンクの目的・目標を規定したものであり、開発担当者自身のみで評価を行う。実用システム開発フェーズ以降では、その内容がマネージャやユーザの要求を反映したものてあるので、ユーザが主体となって評価を行う必要がある。
［各評価者の特徴的な点検項目］
＊開発担当者　：(＊C1)要求仕様の記述内容が一貫性、完備性の観点て問題がないかを点検すること。
＊ユーザ　　：(＊C2)要求仕様の記述内容がユーザの要求を反映しているかの観点で点検すること。
＊運用担当者　：(＊C3)要求仕様の記述内容が運用面の要求を反映しているかの観点て点検すること。
＊マネージャ　：(＊C4)要求仕様の記述内容が組織の目的・目標を反映しているかの観点で点検すること。
（２）要求仕様に対する達成度評価
要求仕様に対する達成度に関しては、原則として、すへてのフェースてエキスパートシステム開発・利用に関係するすへての人によって実施される必要がある。特に、実用システム開発フェース以降では、ユーザが主体となって、厳密に達成度を評価し、総合的にその達成度を判定する必要がある。
［各評価者の特徴的な点検項目］

＊開発担当者　：(＊C1)個々の要件が完全に達成されているかを点検すること。
＊ユーザ　　：(＊C2)自分が使用したい機能が達成されているかを点検すること。
＊運用担当者　：(＊C3)個々の要件が完全に達成されているかを点検すること。
＊マネージャ　：(＊C4)要求仕様全体に関して、との位達成されたか、そして次のフェースへ移行してよいのかの観点て点検すること。

（３）要求仕様設定手段の確認

要求仕様設定手段に関しては、実用ノステム開発フェースて開発担当者か確認または評価する。

［各評価者の特徴的な点検項目］

＊開発担当者　：(＊C1)その手段によって必要十分な要求か網羅され、かつ記述されるかの観点て点検すること。

３２２５知識ベースの保守性評価

知識ベースの保守性の評価に関しては、実用ンステム開発フェースては開発担当者が、運用テストおよび運用・洗練化のフェースでは実際に保守をすることになる運用担当者とユーザとが、それぞれ実施する必要がある。

（１）変更箇所発見の容易性評価

知識ベースに対する変更箇所発見の容易性に関しては、実用ノステム開発フェーズで開発担当者が評価を行う必要がある。

［各評価者の特徴的な点検項目］

＊開発担当者　　(＊C1)例えば、モジュール分割が適切かを点検すること。

（２）知識ベースの理解性評価

知識ベースの理解性に関しては、実用ノステム開発フェースて開発担当者か評価を行う必要がある。

［各評価者の特徴的な点検項目］

＊開発担当者　：(＊C1)知識ベースが知識ベースコーディンク標準に準拠して作成されているかを点検すること。

（３）変更箇所の局所性評価

知識ベースに対する変更箇所の局所性に関しては、実用ノステム開発フェースで開発担当者が評価を行う必要がある。

［各評価者の特徴的な点検項目］

＊開発担当者　　(＊C1)同し項目が複数の個所に重複して記述されていないかを点検すること。

（４）知識ベースの統一性評価

知識ベースの統一性に関しては、実用ノステム開発フェースて開発担当者か評価を行う必要がある。

［各評価者の特徴的な点検項目］

＊開発担当者　：(＊C1)例えば、知識ベースが一貫した方針に基ついて作成されているかを点検すること。

（５）品質安定性の確認方法の確認

品質安定性の確認方法に関しては、開発担当者だけてなく、保守後の品質の安定性を確認する責任のある運用担当者もまた評価する必要がある。

［各評価者の特徴的な点検項目］

＊開発担当者　・(＊C1)品質安定性の確認方法がとの位有効てあるか、との位簡単

であるか、どの位時間がかかるかの観点て点検すること。

＊運用担当者　・(*C1)(同　上)

(6)保守マニュアルの整備状況の確認

保守マニュアルの整備状況に関しては、開発担当者だけてなく、実際に保守を行う運用担当者とユーザもまた確認または評価する必要がある。

[各評価者の特徴的な点検項目]

＊開発担当者　：(*C1)保守マニュアルを完備性、一貫性、有効性の観点で点検すること。

＊ユーザ　：(*C2)保守マニュアルがどの位読み易く、理解しやすく、有効であるかの観点て点検すること。

＊運用担当者　：(*C3)保守マニュアルを完備性、一貫性、有効性の観点て点検すること。

(7)保守用ノールの整備状況の確認

保守用ソールの整備状況に関しては、開発担当者だけてなく、実際に保守を行う運用担当者とユーザもまた確認または評価する必要がある。

[各評価者の特徴的な点検項目]

＊開発担当者　：(*C1)保守用ノールが提供している機能すへてが完全てあるかを点検すること。

＊ユーザ　・(*C2)保守用ノールを有用性、操作性の観点て点検すること。

＊運用担当者　・(*C3)保守用ノールの種類・機能が十分てあるかを点検すること。

3 2 2 6 ヒューマンインタフェースの妥当性評価

ヒューマンインタフェースの妥当性に関しては、その良否が利用者の作業効率を決定付けるので、利用者であるユーザや運用担当者が主体となって評価を行う必要がある。

(1)利用時のインタフェース評価

利用時のインタフェースに関しては、ユーザが主体となって、その使い勝手なとをあらゆる角度から徹底的に評価する必要がある。

[各評価者の特徴的な点検項目]

＊開発担当者　・(*C1)利用時のインタフェースが提供している機能すへてか完全てあるかを点検すること。

＊ユーザ　・(*C2)利用時のインタフェースを操作性の観点て点検すること。

＊運用担当者　(*C3)利用時のインタフェースが提供している機能すへてが完全てあるかを点検すること。

(2)保守時のインタフェース評価

保守時のインタフェースに関しては、保守者てある運用担当者とユーザとが今後発生する保守作業を想定して評価する必要がある。

[各評価者の特徴的な点検項目]

＊開発担当者　：(*C1)保守時のインタフェースが提供している機能すへてが完全てあるかを点検すること。

＊ユーザ　：(*C2)保守がどの位効率的にてきるかの観点て点検すること。

＊運用担当者　・(*C3)保守時のインタフェースが提供している機能すへてか完全であるかを点検すること。

3 2 2.7 システムの総合評価

システムの総合評価に関しては、開発・利用に関係するすべての人が、それぞれの立場で実施する必要がある。

（1）テーマ選定の妥当性評価

テーマ選定の妥当性に関しては、調査・計画およびプロトタイピングのフェーズで、ユーザやマネージャは真のユーザニーズを反映したテーマかの観点で、そして開発担当者や専門家はエキスパートシステム化に適したテーマであるかの観点で、それぞれ評価する必要がある。

［各評価者の特徴的な点検項目］

＊開発担当者　：(＊C1)そのテーマがエキスパートシステム化に適しているかの観点で点検すること。

＊専門家　　：(＊C2)そのエキスパートシステム開発にどれ位協力できるかの観点で点検すること。

＊第3者専門家：(＊C3)その領域がどれ位解明され、どれ位の知識が既に蓄積されているかの観点で点検すること。

＊マネージャ　：(＊C4)そのテーマが組織の目的・目標を達成するものであるかの観点で点検すること。

（2）利用者の満足度評価

利用者の満足度に関しては、マネージャやユーザがあらゆる角度から総合的に継続的に評価を行う必要がある。

［各評価者の特徴的な点検項目］

＊ユーザ　　：(＊C1)自分の仕事をするのに、そのエキスパートシステムかどの位有効であるか、そして生産性や質の向上にどれ位寄与するかの観点で点検すること。

＊運用担当者　：(＊C2)そのエキスパートシステムの利用度の向上の観点で点検すること。

＊マネージャ　：(＊C3)そのエキスパートシステムを使用することにより、自組織での生産性や質がどれ位向上するかの観点で点検すること。

（3）統合性の評価

統合性に関しては、実用システム開発フェーズ以降で、開発担当者は技術的な観点で、ユーザは利用面での評価を行う必要がある。

［各評価者の特徴的な点検項目］

＊開発担当者　：(＊C1)要求仕様どうりに他システムとのインタフェースが適切に行われているかどうかを点検すること。

＊ユーザ　　：(＊C2)システム全体として自分が使用したい機能を満足しているかの観点で点検すること。

＊運用担当者　：(＊C3)他システムと結合した状態でどの位うまく運用できるかの観点で点検すること。

（4）成長可能性の評価

成長可能性に関しては、エキスパートシステム開発・利用に関係するすべての人が将来性を含めて評価する必要がある。

［各評価者の特徴的な点検項目］

＊開発担当者　：(＊C1)最大能力まで成長させるのに、あとどの位の作業量（人・日）が必要かの観点で点検すること。

＊専門家　　：(＊C2)あととの位の知識を提供出来るか、その結果、との位の能力向上が期待出来るかの観点て点検すること。
＊第３者専門家：(＊C3)あととの位の知識がその領域にあるかとうかの観点て点検すること。
＊マネージャ　：(＊C4)投資を継続して、投資効果があるかの観点て点検すること。

（５）開発／保守用トキュメントの評価

開発／保守用トキュメントに関しては、開発担当者自身て十分にレヒューし、評価する必要がある。運用担当者も保守に必要な情報が十分に盛り込まれているかの観点で評価する必要がある。

［各評価者の特徴的な点検項目］
＊開発担当者　：(＊C1)そのトキュメントがトキュメント標準に準拠しているかを点検すること。
＊運用担当者　：(＊C2)そのトキュメントが有効であるか、理解しやすいかの観点で点検すること。
＊マネージャ　：(＊C3)トキュメントが完備しているかの観点て点検すること。

（６）開発方法の適切さの確認

開発方法の適切さに関しては、開発作業の効率・質・コストの観点て、開発担当者とマネージャとが協力して総合的に確認または評価する必要がある。

［各評価者の特徴的な点検項目］
＊開発担当者　：(＊C1)その開発方法を採用することにより、開発作業の効率・質・コストの観点で、どの程度向上したかを点検すること。
＊マネージャ　：(＊C1)（同　上）

（７）ツール・言語選択の適切さの確認

ツール・言語選択の適切さに関しては、開発作業の効率・質・コストの観点て、開発担当者とマネージャとが協力して総合的に確認または評価する必要がある。

［各評価者の特徴的な点検項目］
＊開発担当者　：(＊C1)そのツール・言語方法を採用することにより、開発作業の効率・質の観点で、どの程度向上したかを点検すること。
＊マネージャ　：(＊C1)（同　上）

（８）システム開発の生産性評価

システム開発の生産性に関しては、開発担当のマネージャの最大の関心事の一つであるので、開発担当者とマネージャとが協力して、できるだけ厳密に定量的に評価する必要がある。

［各評価者の特徴的な点検項目］
＊開発担当者　：(＊C1)エキスパートシステム自体の生産性およびエキスパートンステム化することによって、どの程度ンステム開発の生産性が向上したかを点検すること。
＊マネージャ　：(＊C1)（同　上）

（９）システムのコストパフォーマンス評価

システムのコストパフォーマンスは利用者であるユーザとマネージャの最大の関心事であるので、ユーザ、マネージャ、開発担当者の３者て定量的に評価する必要がある。

［各評価者の特徴的な点検項目］
＊開発担当者　：(＊C1)との位のコストでンステムを開発・完成することが出来たかを点検すること。

＊ユーザ　　：(＊C2)そのエキスパートシステムを使用することにより、自分の仕事の生産性や質がどれ位向上したかをコストに換算して点検すること。

＊運用担当者　：(＊C3)そのエキスパートシステムの運用にとの位のコスト（計算資源、人件費、消耗品代など）がかかるかの観点て点検すること。

＊マネージャ　：(＊C4)トータルコストから見て、投資効果があるかを点検すること。

（10）セキュリティ対策の評価

セキュリティ対策は、マネージャ、開発担当者、運用担当者の３者の責任範囲であるので、その３者てセキュリティ対策が十分てあるかを徹底的にテストし、評価する必要がある。

［各評価者の特徴的な点検項目］

＊開発担当者　：(＊C1)要求仕様とうりにセキュリティ対策機能か実現されているかを点検すること。

＊運用担当者　：(＊C2)セキュリティ対策機能がとの位有効て、運用にとれ位の負荷がかかるかを点検すること。

＊マネージャ　：(＊C3)セキュリティ対策機能が費用対効果面から見てとの位有効であるかを点検すること。

3.3 ドキュメント別評価項目

エキスパートシステムの評価を実施するとき、関係者間での質疑応答だけて評価できる場合もあるが、最終的には、その評価対象となる何らかの目に見える形のドキュメント（成果物としてのプログラムリスト等も含む）がないと評価は困難になる。例えば、知識ベースの正当性評価を実施する場合、各開発フェースで作られる知識分析書などをレビューすることにより実質的な評価が可能となる。従って、レビューする観点から、各開発フェーズて作成されるトキュメントことに評価項目をまとめ、特徴的な点検項目／測定項目について列挙する。

3.3.1 ライフサイクル・ドキュメント別による重点評価項目

ライフサイクルの各開発フェーズで作成されるトキュメントを横軸にした重点評価項目の対応表を表3.3-1 に示す。この表中で、丸印をつけて対応づけた項目（縦軸）が各ドキュメントの重点的な評価項目である。開発フェーズごとに作成されるドキュメントは、エキスパートンステムのデザインレビューや評価を実施する場合のガイドラインとなる。

表3.3-1 に基づき、代表的なドキュメントに関する重点評価項目の特徴を以下に説明する。

3.3 1 1 要求仕様書の重点評価項目の特徴

各開発フェーズで作成される５種類の要求仕様書（企画提案書、第Ｉ版要求仕様書、第Ⅱ版要求仕様書、実用版要求仕様書、変更仕様書）に関しては、要求仕様に対する達成度評価に関する下記の評価項目が一番関係が深く、重点項目になる。

（Ａ）要求仕様内容に対する評価
（Ｂ）要求仕様に対する達成度評価
（Ｃ）要求仕様設定手段の評価

次に、システムの総合評価に関する下記の評価項目が重点項目であり、これらの評価項目に留意したレビューや評価が必要てある。

（Ａ）テーマ選定の妥当性評価
（Ｂ）利用者の満足度評価
（Ｃ）統合性の評価
（Ｄ）成長可能性の評価
（Ｅ）開発／保守用トキュメントの評価
（Ｆ）開発方法の適切さの確認
（Ｇ）ツール・言語選択の適切さの確認
（Ｈ）ンステム開発の生産性評価
（Ｉ）システムのコストパフォーマンス評価
（Ｊ）セキュリティ対策の評価

表3　3－1　ライフサイクル別ドキュメント別による重点評価項目（つつく）

分類	評価項目 ＼ 開発フェーズ / ドキュメント	調査・計画		プロトタイピング										実用システム開発							運用テスト	運用先練化			
				実験型					発展型																
		調査報告書	企画提案書	I版要求仕様	I版知識分析	I版問題解決	I版ﾌﾟﾛﾄﾀｲﾌﾟ設計	I版ﾃｽﾄ文書	II版要求仕様	II版知識分析	II版問題解決	II版ﾌﾟﾛﾄﾀｲﾌﾟ設計	II版ﾃｽﾄ文書	実版要求仕様	実版知識分析	実版問題解決	実版ﾌﾟﾛﾄﾀｲﾌﾟ設計	実版ﾃｽﾄ文書	運用ﾏﾆｭｱﾙ	保守ﾏﾆｭｱﾙ	テスト文書	変更要求仕様	運用実績書	保守報告書	改訂版ドキュメント
知識ベースの妥当性評価	(A)知識ベースの正当性評価 [P]									○			○		○			○			○		○		
	(B)知識ベースの完備性評価 [P]														○			○			○		○		
	(C)知識ベースの一貫性評価 [P]									○			○		○			○			○		○		
	(D)経験的知識を正当化する方法とその確認 [E]									○			○												
	(E)知識表現の選択と適切さの確認 [D]				○			○		○			○		○			○							
	(F)知識獲得方法の選択と適切さの確認 [D]				○			○		○			○												
問題解決・推論方式の妥当性評価	(A)推論処理の妥当性評価 [P]					○	○	○			○	○	○			○	○	○							
	(B)推論時間の妥当性評価 [P]										○	○	○			○	○	○			○		○		
	(C)推論手法の選択と適切さの確認 [D]					○		○			○		○			○		○							
	(D)不確実情報に対する処理技法の適切さの確認 [D]					○	○	○			○	○	○			○	○	○							
	(E)学習機能の適切さの確認 [D]					○		○			○		○			○		○							
結果の妥当性評価	(A)正解の評価 [P]						○	○				○	○				○	○			○		○		
	(B)解の網羅性評価 [P]																	○			○		○		
	(C)解の最適性評価 [P]																	○			○		○		
	(D)解の評価方法の適切さの確認 [E]				○	○				○	○				○	○									
要求仕様に達成する度評価	(A)要求仕様内容に対する評価 [P]		○	○					○					○								○			
	(B)要求仕様に対する達成度評価 [P]			○				○	○				○	○				○			○	○	○		
	(C)要求仕様設定手段の確認 [D]			○					○					○								○			

表3　3－1　ライフサイクル別ドキュメント別による重点評価項目（つつき）

分類	評価項目 ＼ 開発フェース	調査・計画		プロトタイピング										実用システム開発							運用テスト	運用・洗練化			
				実験型					発展型																
	ドキュメント	調査報告書	企画提案書	I版要求仕様	I版知識分析	I版問題解決	I版システム設計	I版テスト文書	II版要求仕様	II版知識分析	II版問題解決	II版システム設計	II版テスト文書	実版要求仕様	実版知識分析	実版問題解決	実版システム設計	実版テスト文書	運用マニュアル	保守マニュアル	テスト文書	変更要求仕様	運用実績書	保守報告書	改訂版ドキュメント
知識ベースの保守性評価	(A)変更箇所発見の容易性評価 [P]											○	○				○	○							
	(B)知識ベースの理解性評価 [P]											○	○				○	○							
	(C)変更箇所の局所性評価 [P]											○	○				○	○							
	(D)知識ベースの統一性評価 [P]											○	○				○	○							
	(E)品質安定性の確認方法の確認 [D]																○	○	○		○	○	○	○	○
	(F)保守マニュアルの整備状況の確認 [D]																			○				○	○
	(G)保守用ノールの整備状況の確認 [D]																			○				○	
HI妥当性評価	(A)利用時のインタフェース評価 [P]											○	○				○	○	○		○		○		
	(B)保守時のインタフェース評価 [P]																○	○		○				○	
システムの総合評価	(A)テーマ選定の妥当性評価 [P]	○	○		○	○				○	○														
	(B)利用者の満足度評価 [P]			○			○	○	○			○	○	○			○	○	○		○		○		
	(C)統合性の評価 [P]	○	○		○	○				○	○														
	(D)成長可能性の評価 [P]								○	○	○	○	○	○	○	○	○	○					○		
	(E)開発／保守用ドキュメントの評価 [P]	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	(F)開発方法の適切さの確認 [D]	○	○		○	○	○			○	○	○			○	○	○								
	(G)ノール 言語選択の適切さの確認[D]	○	○		○	○	○	○		○	○	○	○		○	○	○	○					○		
	(H)システム開発の生産性評価 [O]	○	○						○	○	○	○		○	○	○	○								
	(I)システムのコストパフォーマンス評価 [O]	○	○						○					○									○		
	(J)セキュリティ対策の評価 [O]	○	○											○	○	○	○		○				○		

3 3 1.2 知識分析書の重点評価項目の特徴

各開発フェーズで作成される３種類の知識分析書（第Ｉ版知識分析書、第Ⅱ版知識分析書、実用版知識分析書）に関しては、知識ベースの妥当性評価に関する下記の評価項目が一番関係が深く、重点項目になる。

（Ａ）知識ベースの正当性評価
（Ｂ）知識ベースの完備性評価
（Ｃ）知識ベースの一貫性評価
（Ｄ）経験的知識を正当化する方法とその確認
（Ｅ）知識表現の選択と適切さの確認
（Ｆ）知識獲得方法の選択と適切さの確認

次に、システムの総合評価に関する下記の評価項目が重点項目てあり、これらの評価項目に留意したレビューや評価が必要てある。

（Ａ）テーマ選定の妥当性評価
（Ｂ）利用者の満足度評価
（Ｃ）統合性の評価
（Ｄ）成長可能性の評価
（Ｅ）開発／保守用トキュメントの評価
（Ｆ）開発方法の適切さの確認
（Ｇ）ツール・言語選択の適切さの確認
（Ｈ）システム開発の生産性評価
（Ｉ）ンステムのコストパフォーマンス評価
（Ｊ）セキュリティ対策の評価

3 3 2 点検項目・測定項目

エキスパートシステムの開発には、トキュメントとして要求仕様書と知識分析書が少なくても必要である。ここでは、３３１項で示した重点評価項目を、更に詳細化した項目としての点検項目と測定項目について、各トキュメントの記述内容とともに以下に重点的に列挙して示す。これらの点検項目・測定項目は、各トキュメントのデザインレビューを実施する場合に必要になるチェックリストに相当する。ただし、以下に列挙した項目は、ガイドラインとして一例を示したものであり、実際のデザインレヒューでは、これらを参考にした具体的なチェックリストの作成が必要になる。

3 3 2 1 要求仕様書

（１） 企画提案書（第０版要求仕様書）

調査・計画フェーズで作成される企画提案書には主として以下に示す内容が記述される。

a. システム化の目的と範囲
b. プロジェクト要件
　　開発費、納期、要員計画
c. 実現したい機能
d. 導入効果

(1-1) 点検項目・測定項目

記述内容に応じた主な点検項目・測定項目を以下に列挙する。

a. システム化の目的と範囲

《適切な問題》

(C02)経験的知識やノウハウが記述できる問題であること。

(C03)従来のアルゴリズムやデータ構造では設計が難しい問題であること。

(C04)試行錯誤の過程を含む問題であること。

(C05)組合わせが膨大な問題であること。

(C06)ルールは多く複雑だが論理は単純な問題であること。

(C07)過去のデータや結果が一部利用可能な問題であること。

《専門家の存在》

(C08)対象問題の専門家が存在し、その協力が得られること(時間的、心情的に)。

(C09)専門家との円滑なコミュニケーションか可能であること。

(C10)インタヒューの時間が充分に確保できること。

《3 1 4 1 要求仕様内容に対する評価》

(C01)要求仕様の記述内容はユーザの期待と合致していること(正当性)。

- 操作性、応答性、データ入力の簡便性、出力の視覚化、説明機能
- 推論時間、解の正解率、網羅性、最適性
- 他システムとのインタフェースなどの統合性

(C02)要求仕様の記述に誤りや記述漏れ、脱落がないこと(完全性)。

(C03)要求仕様全体を通して矛盾がないこと(一貫性)。

(C04)要求仕様を誤りなく読者に理解させることが出来ること(伝達性)。

(C05)関連する文章内の各要素との照合が容易に出来ること(連絡性)。

(C06)要求仕様に合致する製品を製作し、実現できること(実現可能性)。

b. プロジェクト要件

(C01)体系的な開発方法であること。

(C02)全ライフサイクルを網羅した開発方法であること。

(C03)プロトタイピングに適した開発方法であること。

(C04)その問題に適した開発方法であること。

c. 実現したい機能

(*C1)実現したい機能についての洗いたしとその効果について検討すること。

d. 導入効果

(*C2)計画の段階で、エキスパートンステム導入により得られると期待される利点と、発生が予想される弊害を検討してあること。

(2) 第I版要求仕様書

実験型プロトタイピングフェーズで作成される第I版要求仕様書には主として以下に示す内容が記述される。

a. システム化の目的と範囲(このフェースにおける)

b. プロジェクト要件

c. 機能要件

(2-1) 点検項目・測定項目
記述内容に応じた主な点検項目・測定項目を以下に列挙する。

a. システム化の目的と範囲
(*C3)企画提案書の検討および詳細化を行って、第Ⅰ版の要求仕様書を作成してあること。
《3.1.4.1 要求仕様内容に対する評価》
(C01)要求仕様の記述内容はユーザの期待と合致していること(正当性)。
- 操作性、応答性、データ入力の簡便性、出力の視覚化、説明機能
- 推論時間、解の正解率、網羅性、最適性
- 他システムとのインタフェースなどの統合性

(C02)要求仕様の記述に誤りや記述漏れ、脱落がないこと(完全性)。
(C03)要求仕様全体を通して矛盾がないこと(一貫性)。
(C04)要求仕様を誤りなく読者に理解させることが出来ること(伝達性)。
(C05)関連する文章内の各要素との照合が容易に出来ること(連絡性)。
(C06)要求仕様に合致する製品を製作し、実現できること(実現可能性)。

b. プロジェクト要件
《プロトタイピング》
(C14)開発の方法論として、プロトタイピング手法が採用てきること。
(C15)従来のプログラムであると、条件が頻繁に変わるためその都度の作り直しはかなわない問題であること。
(C16)システムは逐次的に成長発展が可能であること。
(C17)開発に時間とコストがかけられる問題で、当初は実験システムとしての性格を持つ問題であること。

c. 機能要件
(*C4)当面実現すべき機能についての洗いだしとその効果について検討すること。

(3) 第Ⅱ版要求仕様書
発展型プロトタイピングフェーズで作成される第Ⅱ版要求仕様書には主として以下に示す内容が記述される。
a. システム化の目的と範囲(このフェーズにおける)
b. プロジェクト要件
c. 機能要件
d. 性能/品質要件
e. 導入効果

(3-1) 点検項目・測定項目
記述内容に応じた主な点検・測定項目を以下に列挙する。

a．システム化の目的と範囲

(*C5)実験型プロトタイピングを受けて第Ⅱ版要求仕様書の詳細化を行うこと。

《3 1 5 1　要求仕様内容に対する評価》

(C01)要求仕様の記述内容はユーザの期待と合致していること（正当性）。

- 操作性、応答性、データ入力の簡便性、出力の視覚化、説明機能
- 推論時間、解の正解率、網羅性、最適性
- 他システムとのインタフェースなどの統合性

(C02)要求仕様の記述に誤りや記述漏れ、脱落がないこと（完全性）。

(C03)要求仕様全体を通して矛盾がないこと（一貫性）。

(C04)要求仕様を誤りなく読者に理解させることが出来ること（伝達性）。

(C05)関連する文章内の各要素との照合が容易に出来ること（連絡性）。

(C06)要求仕様に合致する製品を製作し、実現できること（実現可能性）。

b．プロジェクト要件

(C06)開発マシンと実行マシンとは同一である必要はないが、開発環境は実用化レベルへの発展を考慮にいれて準備されていること。

(*C6)実用システムの開発（改善、または再開発）をすべきかどうかの検討をすること。

c．機能要件

(*C7)当面実現すべき機能についての洗いだしとその効果について検討すること。

(C05)欲しい機能が実現され、提供されていること。

(C06)操作が簡単であること。

(C10)出力結果が視覚化されていること。

(C12)説明機能があり、分かりやすいこと。

d．性能／品質要件

(*C8)機能要件ばかりでなく、性能要件も加味した要求が達せられること。

e．導入効果

(C01)知識提供の専門家、開発担当者、意思決定するマネジャ、第3者の専門家等からなるレビュー会議において、エキスパートシステムの成長可能性について充分検討すること。

(C02)特に、プロトタイピングシステムの段階において、エキスパートシステムの成長の可能性について充分レビューすること。

(C03)現時点でのエキスパートシステムの能力は専門家と比べて幼稚であり、劣っていると評価された場合でも、エキスパートシステムの成長の可能性を考慮して総合判断をすること。

（４） 実用版要求仕様書

実用システム開発フェーズて作成される実用版要求仕様書には主として以下に示す内容が記述される。

a. システム化の目的と範囲

b. プロジェクト要件

開発費、納期、要員、トキュメント

c. 機能要件

要求される機能、入力データの範囲、出力結果

d. 性能／品質要件

問題解決能力、応答時間、正確性、保守性、使用容易性

e. ユーザ要件／導入効果

導入効果、ユーザインタフェース

f. 環境要件

資源の制約、ハートウェアの制約

(4-1) 点検項目・測定項目

記述内容に応じた主な点検項目・測定項目を以下に列挙する。

a. システム化の目的と範囲

《３１４１ 要求仕様内容に対する評価》

(C01)要求仕様の記述内容はユーザの期待と合致していること（正当性）。

- 操作性、応答性、データ入力の簡便性、出力の視覚化、説明機能
- 推論時間、解の正解率、網羅性、最適性
- 他システムとのインタフェースなどの統合性

(C02)要求仕様の記述に誤りや記述漏れ、脱落がないこと（完全性）。

(C03)要求仕様全体を通して矛盾がないこと（一貫性）。

(C04)要求仕様を誤りなく読者に理解させることが出来ること（伝達性）。

(C05)関連する文章内の各要素との照合が容易に出来ること（連絡性）。

(C06)要求仕様に合致する製品を製作し、実現てきること（実現可能性）。

b. プロジェクト要件

＜改善の場合＞

(*C9)問題解決能力の強化、ヒューマンインタフェースの強化、性能の向上、運用環境への移行（保守用ツールの検討なと）を考慮して、プロトタイプからのスムーズな移行を行うこと。

＜再開発の場合＞

(*Ca)プロトタイプを破棄した理由を明確にして、再開発を行うこと。

《トキュメント全体》

(C01)必要なドキュメントが揃っていること（変更等に係わるトキュメントも含む）

(C02)トキュメントはバージョンアップされていること。

(C03)運用フェーズになってもバージョンアップのルーチンは維持てきること。

(C04)トキュメントの記載は充分てあること。

(C05)トキュメント形式が標準にあっていること。

- 開発用トキュメントは、文書化基準に準拠して作成されているか。
- 保守用トキュメントは、文書化基準に準拠して作成されているか。

(C06)トキュメント内容は、矛盾がなく意味的に正しいこと。
(C07)ドキュメントは、分かりやすくかつ誤解がないように書かれてあること。
(C08)トキュメントをレビューすることにより、そのエキスパートシステムの機能、性能の妥当性及び開発状の問題点等が充分指摘てきること。
《ワークシート関連》
(C09)仕様書や設計書に書くへき事項が網羅的て、かつ適切てあること。
(C10)ワークシート（ＷＳ）が体系化され、完備していると。
(C11)ＷＳの記入すへき項目が必要充分で、かつ適切てあること。
(C12)ＷＳの記入項目が記入しやすいこと。
(C13)ＷＳの記入例が豊富であること。
(C14)ＷＳに対するチェックリストが豊富であること。
(C15)ＷＳが読み易く、レビューしやすいこと。
《ソフトウェアの機能》
(C16)今後、マニュアルの整備にともない使いやすくするために、以下の機能を検討すること。
- オンラインマニュアル
- 多インデックス検索機能
- ＨＥＬＰ機能との一体化
- 知的チュートリアル

c．機能要件
(C05)欲しい機能が実現され、提供されていること。
(C06)操作が簡単であること。
(C08)入力データの準備に時間がかからないこと。
(C09)必要最小限のデータを入力すれば済むようにすること。
(C10)出力結果が視覚化されていること。
(C11)出力結果の信頼性が高いこと。
(C12)説明機能があり、分かりやすいこと。
(C13)学習機能があり、操作性が向上すること。
(C14)再開機能があり、利用を中断することが出来ること。

d．性能／品質要件
(*Cb)性能と品質基準はユーザの要件を十分満たしていること。

e．ユーザ要件／導入効果
(C15)ユーザの改善要望を機器、改善・機能拡張がタイムリーに行われること。
(C16)教育プログラムが充実し、随時実施されていること。
(C17)ユーザマニュアルが充実し、分かりやすいこと。

f．環境要件
(C01)ツールのサービス、サポート体制、使用契約、価格が要求条件に適合していること。
(C02)開発から実運用までの一貫性を考慮して、ツール・言語の選択を検討すること。

(C01)セキュリティ対策には、技術的対策、運用上の対策、設備上の対策などがあるが、全ての角度から充分な対策を講しること。
(C02)特に、社会システムやその他のシリアスな対象に用いられるエキスパートンステムの場合は、まず起こり得るリスクを事前に充分に検討すること。
例えば、解答が誤った場合、利用者の財産、生命に危具を与え、取り返しのつかない事態を招く恐れもある。
(C03)知識、ノウハウに対する知的所有権、侵害を受けた場合の法的権利などについても事前に明確にすること。
(C04)利用者（社内／社外／不特定多数）の範囲の検討を行うこと。
(C05)情報の開示レベルを検討すること。
(C06)知識ベースは理解性が高いために、逆に内容が漏洩したり改ざんされたりする危険性が高いことに注意すること。
(C07)設計ノウハウの隠蔽（ルールコンパイル等）対策をすること。
(C08)システム自体の実行権および知識ベースへのアクセス権を厳格に設定し、充分なアクセスコントロールを実施すること。
(C09)不正アクセスや改ざんが行われたか否かを検査する手段を設けること。
(C10)ネットワークのデータ伝送路上で充分なセキュリティ対策を講しること。

（５） 変更要求仕様書

運用洗練化フェーズで作成される変更要求仕様書には主として以下に示す内容が記述される。

a. 保守性
b. 変更の容易性
c. 拡張性

(5-1) 点検／測定項目

記述内容に応じた主な点検項目・測定項目を以下に列挙する。

a. 保守性

(*Cc)システムの保守性が高められていること。
(*Cd)システム利用の満足度を把握すること。

b. 変更の容易性

(*Ce)システムの拡張、変更が容易になっていること。

c. 拡張性

(C05)知識ベースはどのあたりまで拡張を予定して設計してあるのかを確認すること。
- 推論エンジン、推論戦略知識、専門知識の分離（保守性）
- 知識修正、追加が容易に行われるエディタ環境の整備

3 3 2.2 知識分析書

プロトタイピング開発フェーズ、実用システム開発フェースて作成される知識分析書には主として以下に示す内容が記述される。

a. 知識および知識源の同定
b. 知識の獲得
c. 知識の分析

（1） 点検項目・測定項目

記述内容に応じた主な点検項目・測定項目を以下に列挙する。

a. 知識および知識源の同定

《3 1 1 1 知識ベースの正当性評価》

(C01)整理された知識は、知識源を遡りその出所を確認すると共に、その知識を用いることの利点、根拠を明らかにしておくこと。

(C02)獲得した知識は、型付け（ファクト、ルール、クラス、実体、関係、構造、行為、制約、資源）を行い、整理すること。
故障木の場合は、ノートとリンクの型付けを行う。

(C03)用いる知識の重要度、詳細度、変化の可能性、使用頻度などの観点から、知識を整理すること。

(C04)以上から、専門家（第3者専門家も含む）に概念レヘル、意味レヘルのレビューを行ってもらうこと。

(C05)その結果、専門家と開発者との間で知識に対する解釈の相違（多犠牲、曖昧性）に関して問題点を洗いだし、それを検討すること。

(C06)特に、曖昧性、信頼性に欠ける知識については、他の専門家の意見を聞き、どの程度信憑性があるのか、又その知識の適用範囲についても検討すること。

(C07)複数の専門家により抽出された知識が異なる場合、知識の解釈が異なる場合、専門家間で合意を得ること。

《3.1.1.4 経験的知識を正当化する方法とその確認》

(C01)知識の獲得に協力した専門家自身の知識の偏りをとう評価するのかは問題であるが、第3者専門家等の意見を入れて充分検討をすること。

(C02)経験的知識は、それ自身客観的に正当化を図ることが困難てあるため、客観的な深い知識に対する資料の整備、その深い知識との比較、一般的に成り立つかどうかの検討などその知識に対して何等かの正当化理由が見当たらないかを検討すること。

(C03)数値データまたは数量化が可能なデータて再現性のある経験的な知識の場合、統計的手法を用いてどの程度の信憑性があるのかを確かめることができるので、その検討を行うこと。

(C04)言葉上似通っている知識でも、その意味する内容が全く異なることがあるので、この場合どう異なるのかを深く追及すること。
例）故障診断で「急激な温度上昇」と「瞬間的な温度上昇」

(C05)過去の実績データがある場合は、そのデータにより経験的知識が裏付けられるのかを検討すること。

(C06)「経験的知識」＝「ローカルな知識」であるから、その適用範囲について充分注意すること。

b. 知識の獲得

《3 1 1.6　知識獲得方法の選択と適切さの確認》

《知識源の妥当性》

(C01)知識を提供する専門家の選定が適切であること。

- 開発協力の得られる知識提供者の専門家が身近にいるか。
- 誰もがその人を専門家と認めているか（一般意見）。
- その専門家の上司または前任者もその人を認めているか。

(C02)参考にした資料が適切であること。

- 古くないか（バージョンアップされているか）。
- 偏りがないか。

(C03)対象とする専門領域の知識を可能な限り収集して事前に修得しておくこと。
深い知識（原理・原則的知識）、浅い知識（経験的知識）、専門家からの知識、文献・資料に基づく知識など。

《知識獲得過程の妥当性》

(C04)専門家のおかれた環境や立場を勘案してインタビューすること。

(C05)インタビューの戦略が適切であること。

- 事前準備を行ったか（たたき台をもとに行ったか）。
- 視覚的に分かりやすい方法を用いて専門家に確認を取ったか。
- 専門家とのコミュニケーションは充分であったか。
- 本来専門家はXを思っていても、Xに近いX´の程度しか言わないので注意すること。

(C06)診断の知識の場合、仮説（診断結果）、兆候（仮説の発生により生しる出来事や状態）、条件（直接的には仮説の前兆とならないが他の兆候の診断の際に影響を及ぼす状況内の出来事や状態）、リンク（実態間の因果関係の表現に使用）、パス（仮説と兆候を結合する）などに分類して整理すること。

(C07)項目と項目の関係知識を整理したが、また曖昧な関係である、専門家が納得できないときは、専門家自身意識になかった問題解決における重要な矛盾があるため、専門家はこれらに気づいて検討すること。

(C08)問題のタイプに適切な知識獲得支援方法またはノールを使用すること。

- Personal Construct Theoryに基づく格子(grid)方法論を用いた自動インタビュー
- ルールの不備な部分の検索を行ない、より正確なルールを生成するため専門家に問い合わせる。

(C09)知識の精粗のレベルを統一すること、不足している知識は何かが明確にわかること。また獲得が困難であるところについて、今後の知識獲得手段につき充分検討すること。

(C10)ＫＥとの協力を必要とする場合、専門家とＫＥとの役割か適切であること。

(C11)新しい状況の場合、それに対応できる専門家はいないので、この場合の知識獲得をどうすべきかも踏まえて検討すること。

(C12)テストケースの収集も知識獲得と合わせて、プロジェクトの早期の段階から始めておくこと。
専門家が知識ベースを信頼できなければ、実用化にはならないため、専門家が納得できるように、テストに大いに参加してもれうことが重要である。

c．知識の分析

《3 1 1 2　知識ベースの完備性評価》

(C01)知識の分解レベルの詳細度がアンバランスになっていないこと。

例えば、車の故障診断の場合、電気系が冷却系よりも詳細に知識分析書に記述されている場合、本当にそれで妥当なのか。

(C02)ある属性だけの分解レベルが細分化されて、アンバランスになっていないこと。

細分化が不備てある属性は再検討する必要がある。

(C03)テスト知識の欠落をチェックすること。

テスト知識とは、ＡｏｒＢのとき、ＡとＢのいずれかてあるかを判別する知識で、これは専門家にインタビューしないと出てこない知識てある。

(C04)取り扱う対象の知識を充分に完備することは困難なので、むしろ何処まで完備したのか、今後どう段階的に発展するのか等を充分把握すること。

(C05)ルールの導出レベル、フレームの階層構造のレヘルがアンバランスになっていないか、構造ツリー等を表示してチェックすること。

《3.1 1 3　知識ベースの一貫性評価》

(C01)対象モデルが外部環境に耐えられること。

(C02)同一対象に対して、複数の専門家の知識を採用した場合、それらの知識に関して整合性が取れていること。

(C03)対象とする問題の範囲を変更した場合、それに関係する知識を適切に変更すること。

(C04)応用分野の特性が整理され、その知識の階層性、継承性、断片性などの特徴を整理すること。

３．４　利用形態別評価項目

エキスパートシステムに対する要求は、その利用形態により異なる。そこで、本項では、利用形態別の評価について述べる。表３４-1 は、利用形態毎の重点評価項目を示す。以下、本表に沿って説明する。

３４１　利用形態の分類

エキスパートシステムの利用形態とは、エキスパートシステムの稼働環境における位置付け、および、環境とのインタフェースである。そこで、本項では、利用形態を分類するための視点として、エキスパートンステムの行なう業務、その業務の遂行におけるエキスパートシステムの役割、利用者の性質、利用者とエキスパートシステムとのインタフェースの性質、他のシステムとの連動を挙げる。このうち、エキスパートシステムの行なう業務については、問題のタイプ別評価項目（３　５節）で述べる。以下、それ以外の視点からの分類について述へる。

（１）業務の遂行におけるエキスパートシステムの役割
専門的な判断の主な部分を人間が行ない、エキスパートンステムがそれを支援する場合（専門家の支援）と、専門的な判断を主にエキスパートンステムが行なう場合（専門家の代行）に分類できる。

（２）利用者の性質（注）
(2-1) 利用者の専門度
利用者が当該業務の専門家てある場合と、そうでない場合に分類できる。
別の見方として、
(2-2) 利用頻度
利用者がそのエキスパートシステムを定常的に利用する人てある場合と、たまに利用する人である場合に分類できる。
（注）ここでは、利用者とは、エキスパートシステムを最も直接的に利用する人間と定義する。従って、エキスパートンステムを他のンステムが利用し、そのシステムを人間が利用する場合、人間が利用者となる。また、エキスパートシステムをある人間が直接利用し、その人間を介して他の人間が間接的にエキスパートシステムを利用する場合、前者の人間が利用者となる。
（３）利用者とのインタフェースの性質
対話的な利用とそうでない場合に分類できる。
（４）他のシステムとの連動
連動する場合とそうでない場合に分類できる。

3 4.2 利用形態別評価項目

上記の視点のうち（１）の分類に基つく評価項目の抽出と、(2-1) の分類に基づく評価項目の抽出は、互いに関連がある。例えば、エキスパートンステムの役割が専門家の代行で、利用者が専門家でない場合を考える。この場合、エキスパートシステムの解を確認する専門家が利用環境にいないことになるので、信頼性の高いエキスパートシステムでないと利用できない。従って、解の正解率評価が重要になる。これに対し、エキスパートンステムの役割が同しく専門家の代行であっても、利用者が専門家の場合には、エキスパートシステムの解を確認することができるので、上記の場合に比へると、正解率評価は重要てない。このように、エキスパートシステムの役割だけを考慮しても評価項目を抽出することはできず、エキスパートンステムの役割と利用者の性質の組合せを考慮する必要がある。そこで、（１）の分類と(2-1) の分類の組合せを考え、組合せのそれぞれについて評価項目を抽出する。

（１）、(2-1) 以外の視点については、それぞれ独立に評価項目を抽出てきる。そこで、各視点における分類毎に評価項目を抽出する。

（１）エキスパートシステムの役割と利用者の専門度から見た評価項目の抽出

(1-1) 役割＝専門家の支援、利用者＝専門家

専門家が、思考のツールとしてエキスパートシステムを利用する場合である。ここては、専門家の見落としや誤りをエキスパートシステムに指摘してもらいたいという要求がある。また、エキスパートンステムの推論過程が利用者となる専門家の思考過程に対応していることが望ましい。さらに、利用者が専門家てあり、エキスパートンステムの誤りや機能の不足を見出すことがてきるので、エキスパートンステムの機能を自分で拡充したいという要求がある。

以上から、この利用形態において特に重要な評価項目として、以下が上げられる。

①解の網羅性評価（３　１節における項目番号は３－２）
(*C1)利用者が許容できる応答時間内て、できるだけ多くの解を求められること。
②知識ベースの完備性評価（１－２）
(*C1)網羅的に多くの解を求めるために、各々の問題や場合において適用可能な知識をたとえ重複していてもできるだけ多く持つこと。
③推論手法の選択と適切さの確認（２－３）
(*C1)推論手法選択の基になる専門家モデルが、知識提供の専門家に特有ではなく、他の専門家にも受け入れられるものであること。
④知識表現の選択と適切さの確認（１－５）
(*C1)知識表現選択の基になる専門家モデルが、知識提供の専門家に特有ではなく、他の専門家にも受け入れられるものてあること。
⑤経験的知識を正当化する方法とその確認（１－４）
(C01)［知識の偏り］経験的知識が知識提供の専門家に特有のものにならないように、第３者専門家あるいは利用する専門家の意見を取り入れること。
⑥正解の評価（３－１）
(C13)［問題解決の仕方、流儀］正解の定義が知識提供の専門家に特有のものに

表3　4－1　利用形態別による重点評価項目（つづく）

分類	評価項目 ＼ 利用形態	専門家の支援	専門家の代行		利用頻度		インタフェース		他システム	
		専門家の利用	専門家の利用	非専門家の利用	定常的	非定常的	対話的	非対話的	連動	単独利用
1 知識ベースの妥当性評価	1-1知識ベースの正当性評価 [P]			○					○	
	1-2知識ベースの完備性評価 [P]	○								
	1-3知識ベースの一貫性評価 [P]									
	1-4経験的知識を正当化する方法とその確認 [E]	○	○							
	1-5知識表現の選択と適切さの確認 [D]	○								
	1-6知識獲得方法の選択と適切さの確認 [D]									
2 問題解決推論方式の妥当性評価	2-1推論処理の妥当性評価 [P]								○	
	2-2推論時間の妥当性評価 [P]				○		○		○	
	2-3推論手法の選択と適切さの確認 [D]	○					○			
	2-4不確実情報に対する処理技法の適切さの確認 [D]									
	2-5学習機能の適切さの確認 [D]				○					
3 結果の妥当性評価	3-1正解の評価 [P]	○	○	○					○	
	3-2解の網羅性評価 [P]	○								
	3-3解の最適性評価 [P]			○						
	3-4解の評価方法の適切さの確認 [E]									
4 要求仕様に対する達成度評価	4-1要求仕様内容に対する評価 [P]									
	4-2要求仕様に対する達成度評価 [P]									
	4-3要求仕様設定手段の評価 [D]									

表3　4－1　利用形態別による重点評価項目（つつき）

分類	評価　目	専門家の支援	専門家の代行		利用頻度		インタフェース		他システム	
	利用形態	専門家の利用	専門家の利用	非専門家の利用	定常的	非定常的	対話的	非対話的	連動	単独利用
5 知識ベースの保守性評価	5-1変更箇所発見の容易性評価 [P]				○					
	5-2知識ベースの理解性評価 [P]	○	○		○					
	5-3変更箇所の局所性評価 [P]				○					
	5-4知識ベースの統一性評価 [P]				○					
	5-5品質安定性の確認方法の確認 [D]				○					
	5-6保守マニュアルの整備状況の確認 [D]	○	○		○					
	5-7保守用ノールの整備状況の確認 [D]	○	○		○					
6 HI妥当性の評価	6-1利用時のインタフェース評価 [P]	○	○		○	○	○			
	6-2保守時のインタフェース評価 [P]	○	○		○					
7 システムの総合評価	7-1テーマ選定の妥当性評価 [P]									
	7-2利用者の満足度評価 [P]				○					
	7-3統合性の評価 [P]								○	
	7-4成長可能性の評価 [P]									
	7-5開発／保守用ドキュメントの評価 [P]									
	7-6開発方法の適切さの確認 [D]									
	7-7ノール 言語選択の適切さの確認 [D]	○	○		○		○		○	
	7-8システム開発の生産性評価 [O]									
	7-9システムのコストパフォーマンス評価 [O]	○	○	○						
	7-10セキュリティ対策の評価 [O]									

ならないように、第３者専門家あるいは利用する専門家の意見を取り入れること。

⑦知識ベースの理解性評価（５－２）

（C01）［専門家による保守］専門家による保守を想定したモジュールは、プログラミングの知識がなくても理解できるようになっていること。

⑧保守時のインタフェース評価（６－２）

（*C1）専門家自身による知識ベースの保守が可能であること。すなわち、プログラミングの知識がなくても保守できること。

（C07）［内部表現への変換］専門家の知識表現によって入力した知識をシステム固有の表現に変換する機能を持つこと。

⑨保守マニュアルの整備状況の確認（５－６）

（*C1）マニュアルの内容、表現が、専門家にとって理解しやすいこと。

⑩保守用ツールの整備状況の確認（５－７）

（C02）［専門家による保守］保守用ツールは、プログラミングの知識がなくても操作できること。

⑪利用時のインタフェース評価（６－１）

（*C1）専門家にとって自明な情報は表示しないこと。

⑫システムのコストパーフォーマンス評価（７－９）

（*C1）専門家の作業効率の向上、誤りの減少を評価すること。

⑬ツール・言語選択の適切さの確認（７－７）

（*C1）上記の推論、保守、インタフェースに関する要求を達成するに適しているかどうかを確認すること。

（1-2）役割＝専門家の支援、利用者＝非専門家

この形態はありえない。

（1-3）役割＝専門家の代行、利用者＝専門家

エキスパートシステムが専門家の業務を代行し、他の専門家がその結果を利用する場合である。（1-1）と同様の理由により、利用者がエキスパートンステムの誤りや機能の不足を見出すことができるので、エキスパートンステムの機能を自分で拡充したいという要求がある。本利用形態における重要評価項目を以下に上げる。

①知識ベースの理解性評価（５－２）
②保守時のインタフェース評価（６－２）
③保守マニュアルの整備状況の確認（５－６）
④保守用ツールの整備状況の確認（５－７）
⑤経験的知識を正当化する方法とその確認（１－４）
⑥正解の評価（３－１）
⑦利用時のインタフェース評価（６－１）
⑧ツール・言語選択の適切さの確認（７－７）
（以上、上記（１－１）の点検項目と同様）
⑨システムのコストパーフォーマンス評価（７－９）
（*C1）専門家の作業量の削減を評価すること。

(1-4) 役割＝専門家の代行、利用者＝非専門家

利用者が解の正誤を判断できない場合が多いので、正解率の高いシステムでないと利用できない。また、複数の解の中から最適解を選択できない場合が多いので、解のランク付け、あるいは、最適解のみを出力することが必要である。本利用形態における重要評価項目を以下に上げる。

①正解の評価（３－１）

(C05)［要求仕様］要求仕様で記述した正解率以上であること。

②解の最適性評価（３－３）

(*C1)利用者が複数の解の中からの選択に困らないように、解をランク付けしたり、最適解のみを出力することが必要である。そのための最適性評価が可能であること。

③知識ベースの正当性評価（１－１）

高い正解率を保つために、特に、以下の項目が重要である。

(C01)［知識源の確認］知識の出所を確認すること。

(C02)［知識の型付け］獲得した知識は型付けを行い、整理すること。

(C03)［知識の整理］用いる知識の重要度、詳細度、変化の可能性、使用頻度などの観点から知識を整理すること。

(C04)［専門家によるレビュー］以上から、専門家に概念レベル、意味レベルのレビューを行ってもらうこと。

(C05)［問題点の抽出・検討］その結果、専門家と開発者との間で知識の解釈の相違に関して問題点を洗いたし、それを検討すること。

(C06)［知識の信憑性］特に、信頼性に欠ける知識については、他の専門家の意見を聞き、どの程度信憑性があるのか、またその知識の適用範囲についても検討すること。

(C07)［専門家間での合意］複数の専門家の間の知識の不一致を解消するために、専門家間で合意を得ること。

(C08)［ウォークスルー］エキスパートシステムの解決過程と専門家の解決過程とを比較検討して、どの知識が用いられているかを逐一ウォークスルーすること。

(C11)［知識の制御］知識の活性化のタイミングや優先順位が適切であること。

④システムのコストパフォーマンス評価（７－９）

(*C1)専門家の作業量の削減を評価すること。

（２）利用頻度から見た評価項目の抽出

(2-1) 定常的利用者

定常的利用者を想定したシステムでは処理や操作の効率が重要である。さらに、利用者が非定常的である場合に比べ、システムの不備に対する改善要求が厳しいので、システムの保守性や成長性が重要である。本利用形態における重要評価項目を以下に挙げる。

①推論時間の妥当性評価（２－２）

(*C1)利用者の満足できる応答時間であること。

②利用時のインタフェース評価（６－１）

(*C1)情報の表示が簡潔であること。
(C02)［入力の操作性］操作が効率的であること。
③知識ベースの保守性評価（５）
改善要求が厳しいので、短期間で保守できること、すなわち、保守の効率が重要である。そこで、変更箇所発見の容易性、変更箇所の局所性、保守用ノールの整備が重要である。また、保守を繰り返しても、知識ベースの一貫性が失われないためには、知識ベースの統一性が重要である。
④保守時のインタフェース評価（６－２）
知識ベースの保守を行うためのインタフェースが良いこと。特に、
(*C1)操作が効率的であること。
⑤学習機能の適切さの確認（２－５）
(*C1)システムが利用者のくせに適応すること。
(*C2)同し誤りを繰り返さないこと。
⑥利用者の満足度評価（７－２）
(*C1)1、２回利用する場合に比べ、繰返し利用する場合の方が、利用者の要求が厳しい。そこで、定常的利用者を想定したシステムの満足度評価にあたっては、繰り返し利用してもらった後、満足度を調へること。
⑦ツール・言語選択の適切さの確認（７－７）
(*C1)上記の推論、保守、インタフェースに関する要求を達成するに適しているかどうかを確認すること。

(2-2) 非定常的利用者
非定常的利用者を想定したンステムでは操作の分かり易さか重要である。重要評価項目は以下のとおりてある。

①利用時のインタフェース評価（６－１）
(*C1)操作が分かり易いこと。
(C01)［結果の表示］情報の表示が分かり易いこと。
(C04)［ＨＥＬＰ機能］ＨＥＬＰ機能が充実していること。

（３）インタフェースの性質から見た評価項目の抽出

(3-1) 対話的利用
応答が一定時間内てあること、対話の進行が自然てあること、インタフェースの使い勝手がよいことが重要てある。重要評価項目は以下のとおりてある。

①推論時間の妥当性評価（２－２）
(*C1)利用者の満足できる応答時間であること。
②推論手法の選択と適切さの確認（２－３）
(*C1)対話のシナリオが明確化されており、適切てあること。
(*C2)上記ンナリオに沿った推論手法であること。
③利用時のインタフェース評価（６－１）
(C01)［結果の表示］結果の表示や判断・応答の入力補助情報の表示において適切な表現方法が用いられていること。特に、
・文脈の表示方法と表示順序。

(C02)［入力の操作性］入力の操作性が良いこと。特に、
　　・長時間使用しても身体的な苦痛が少ないこと。
(C04)［ＨＥＬＰ機能］ＨＥＬＰ機能の充実とその操作性が良いこと。
　　・文脈による必要最小限のＨＥＬＰ
(C05)［文脈の表示］明確な文脈の表示であること。
(C12)応答時間。
(C13)［ＷＨＹ機能］説明機能、特に、ＷＨＹ型説明機能が充実していること。
(C17)［対話性］冗長な質問を出さないこと、利用者主導の対話制御か可能てあること。
④ツール・言語選択の適切さの確認（７－７）
(*C1)上記の推論、保守、インタフェースに関する要求を達成するに適しているかどうかを評価すること。

(3-2) 非対話的利用
非対話で利用する場合、特有の評価項目はない。

（４）他システムとの連動から見た評価
(4-1) 連動
まず、トータルンステムとしての評価が重要てある。また、エキスパートンステムの誤りが他システムの誤動作を引き起こし、その結果、エキスパートンステムを単独利用した場合以上に大きな障害を引き起こす場合がある。例えば、機械系を制御するエキスパートシステムが誤った場合には、環境に物理的な破壊をもたらす恐れがある。そこて、エキスパートシステムの信頼性が重要である。さらに、他システムとオンラインて連動している場合には、処理速度が他ンステムに合っていなければならない。重要評価項目は以下のとおりである。
①統合性の評価（７－３）
従来型技術による機能との連携や統合性の良さを評価すること。特に、
(*C1)エキスパートンステムの誤りに対するフェイルセーフ機能を備えていること。
②正解の評価（３－１）
(*C1)解の満たすへき最低限の制約を明らかにし、その点に関してテストを厳密に行うことにより、致命的な誤りを防止すること。
③知識ヘースの正当性評価（１－１）
(*C1)エキスパートンステムの想定している問題範囲と連動ンステムのそれとの一致を考慮すること。
④推論処理の妥当性評価（２－１）
(*C1)フェイルセーフの機能を備えていること。
⑤推論時間の妥当性評価（２－２）
(*C1)処理速度が連動ンステムに合っていること。
⑥ツール・言語選択の適切さの確認（７－７）
(C03)［高性能のツール選択］他ンステムに対する呼出しおよび被呼出し機能を備えていること。

(4-2) 単独利用
単独利用する場合、特有の評価項目はない。

3 5　問題のタイプ別評価項目

ここではエキスパートンステムの対象とする問題を合成型問題、解析型問題、複合型問題の3つに分け、問題ごとに評価すべき内容の特徴、必要性について述べる。なお合成型問題は設計型問題と計画型問題に細分し、解析型問題は解釈型問題、診断型問題、制御型問題に細分して検討する。

問題の対象は「ドメイン」と呼ぶ。一方、対象とする問題を分析したとき、その問題を解決するのに必要とする特有な作業の型が見いだせる。これを「タスク」と呼ぶ。ここでは、エキスパートシステムをタスクとトメインの2つの軸、即ちタスクドメインの特徴的な問題としてとらえることにする。

従って、本節では、各問題のタイプに対し、トメインの特徴及びタスクの特徴を整理して述べ、それぞれの問題について、評価項目の重要性の説明、特徴的な点検項目・測定項目に詳細な説明について述へる。3 1 節を参照している点検項目・測定項目は同一番号で対応を取っている。ただし、先頭に＊印のある点検項目・測定項目は、問題のタイプ特有の点検項目・測定項目である。

なお本節で扱う「問題領域」とは、設計、計画など一般的に用いられる用語を使用しているが、その定義は小林らの定義に基づくもので、業務分野に共通する基本的な問題解決タスクの分類に従っている［AIｾﾝﾀ　1989］［Kobayashi 1990］。

以下に各問題について、システムの特性、即ち定義、問題の特徴、基本タスク、問題解決機能（実現手法）を表3 5-1 にして取りまとめた。

《参考文献》

［AIｾﾝﾀ　1989］知的情報処理システムに関する調査研究報告書、（財）日本情報処理開発協会、ICOT-JIPDEC AIｾﾝﾀｰ、1989

［Kobayashi 1990］小林重信、寺野隆雄（編）：知識システムハントブック、オーム社、1990

表3 5-1 システムの特性一覧表

	設計型システム
定義	与えられた要素を決定すべき仕様が構成要素の内部構造を決定すべき仕様を満たすように、構成要素の組合せ、構成要素の最適化の必要性を検討する。
問題の特徴	1. 解空間が大 2. 解析的基本的 3. 構成要素の完全性 4. 検証の効率性 5. 検証の形式性 6. 対話的設計支援 7. 設計段階に応じた支援
基本タスク	1. 構成要素の関連の表現 2. 設計問題の階層的表現 3. 代替案の自動生成 4. 部分テストの評価 5. 上位レベルへの知的後戻り 6. 設計事例の検索と利用 7. 並列的問題解決
問題解決機能	1. トップダウン精密化戦略 2. 拘束最小化戦略 3. 最適化機能 4. 検証機能 5. 仮説推論による知的探索 6. 類推による設計事例の利用 7. 協調型推論

	計画型システム
定義	与えられた目標を実現するために、利用可能な資源（人、金、物など）を時系列データとして組織化することである。
問題の特徴	1. 広い探索空間 2. 多様な評価属性 3. 不確実な環境予測 4. 部分計画間の相互作用 5. 既存計画の再利用 6. 費用と便益のトレードオフ 7. 対話型計画作成支援の要求
基本タスク	1. 計画過程の階層化 2. 組合せ的探索 3. 制約条件の相互作用 4. 環境の予測 5. 上位レベルへの知的後戻り 6. 計画事例の検索と利用 7. 計画評価の多様性
問題解決機能	1. トップダウン精密化戦略 2. 拘束最小化戦略 3. 網羅的探索機能 4. 仮説推論による知的探索 5. 類推による計画事例の利用 6. 計画の多目的評価 7. リスク下での意思決定

	解釈型システム
定義	たしされ定あされ推で測をと観態こて状るしえ通ム与をテをサスけンノ付セ味はて意いしなる析的あ分理器を物測タに計一れデこ
問題の特徴	1. 相関性 2. 欠入 3. 高混 4. のの 5. 欠のデ扱 6. のとモ取 7. 落接ルい 続の利用 タ音 一雑在一技すタ デな存デ理関一 間則のな処にデ 空規り要号象覚 時不誤必信対感
基本タスク	1. 出 2. 抽 3. 完 4. 不 5. 造 6. 態 7 性 特モンノあ不解 徴デススい完釈 のルテテま全の 抽とムムい性多 出の構状さの様 昭合 な定定 全同推理 完ののの処理 不造態の処理
問題解決機能	1. クラス分け 2. ク能 3. 機能 4. る機法 5. よ能価査 6. 機評検論 7. に索の成推論 層検造生性論推 階ル構的実推型 類デ分層確説調 分モ部階不仮協

	診断型システム
定義	生にるがムす障テ定故ス同はンをたび所ま及箇常タ因異一原にデムたて、テれしスさ用ン測利る観をす識る。とき、知あ象とるで対たすと関こ
問題の特徴	1. 利用 2. 識の 3. 知実性 4. 利用 5 利間と コスト 6. 不必要性 7. 時論制御 ルのの ベ知知一た抽診 レ的的デのの的 計験験業測識話 設経経操計知対
基本タスク	1 現表現 2. 的表現 3. 内包 4. 階層的 5. 類層を定 6. 分階決 7. 造タ択同にに の構の選のルル 象ム題の因デデ 現テ問点原モモ 常ス釈測常いい 異ノ解計異浅深
問題解決機能	1. 推論 2. 型推論 3. 動型推論 4. 動性論 5. 駆推論 6. 駆実推型性／ 7. 象標確説調率用 事目不仮協効費 効率性 完全性 全性の調和 完分析 果

	制御型システム
定義	定予よめる予て、予めて状態が遷移する。視のある。監ムでと況スこ状ンるのにえムり加テ通をスた作ンれ操さに
問題の特徴	1. 時間線定御プ時話 依存性 2. 非遅形化精ロ間的 歴所性 3. 安制れ性の度セ性設 履局 4. 実にに応のスと計 る現 5. 実よ答実と信支 よ接点 6. 対 の頼性 7. 要求
基本タスク	1. 現 2. 表の ク 現 3. 性を 内 包 4. ス的 予 測 5. 状態 実 行 6. 優先 ネ 化 7. の省エ ス 8. イダ 処理
問題解決機能	1. 能 2. による予測 3. 機 測 4. タ評 価 5. よ ダ ン 6. 解析 7. 回避

	複合型システム（解析＋合成）
定義	報をのて情結所このらそて、部からは外するも行い推条上るを束み態拘組あ状たを身、のれ解自象らすテ対めたム決満スし、ン析に、を分基要ある。
問題の特徴	1. 対象に関する情報を再属変更 2. 不完全分析性の価値評価 3. 不十な 4. あ存計な 5. 既様 6. 多的 7. 動
基本タスク	1. 対象のモデル化 2. 対象のモデル推定 3. あい状態不完全性）の処理 4. 解釈多様性 5. 推定約制約化 6. 制約結果的利用 7. 組み合わせ探索
問題解決機能	1. モデル類階層プ類 検索機能（類推） 2. 分類階層ノ約条 探索 3. 階生約調 査法 4. ト制ダ件案 細化戦略 5. 集計 整 小化 6. 計画の総 最 法 7. 合評価法（得点法）

3.5.1 設計型システム

3.5.1.1 ドメインの特徴

(1) 設計型システムの定義

設計型問題とは，設計対象のシステムの入力と出力の要求仕様が与えられたとき、これを実現するために構成要素を組み合わせ、かつ各構成要素の内部仕様を決定することであり［Kobayashi 1990］、その問題を解決するシステムを設計型システムという。

(2) 設計型システムの実現手法

次のような専門家の問題解決過程のいくつかの側面に着目してシステムを実現することが多い。

①探索的側面

所望の構成・特性を得るため、結果の候補案を何らかの方法で生成して、解析・検査し、制約条件に合致していれば結果として採用する。エキスパートシステム化では、この候補案の追求を探索問題に帰着させている。

②意思決定的側面

結果の候補案を生成しても、この時点ではその候補案が結果として適切であるかどうかを判定することが不可能・不明であれば（さらに詳細化すれば判定できる、従属関係にあるほかの部分の設計がある程度すすめば判定できる、など）、候補案を仮に選択しておく必要がある。エキスパートシステム化では、この候補案の選択を意思決定の問題に帰着させている。

③再利用的側面

専門家の問題解決過程では過去に経験した成功および失敗の事例は有効に利用されており、新しい設計問題が与えられた場合にも、これと類似の過去の事例を検索し、これを参考にしながら問題解決過程の効率を向上させている。エキスパートシステム化では、これを過去の事例の再利用の問題に帰着させている。

④知識獲得的側面

類似の設計事例を利用できない全く新しい条件の問題においては、専門家は問題解決に有用な知識を自ら獲得していくことが必要になる。エキスパートシステム化では、この知識の獲得の過程を、多数の正例（成功例）や負例（失敗例）の背後に成立する法則・原理・規則を帰納的に導く、あるいは、既に獲得済の知識を利用して新しい事例を説明し、かつこれらの一般化を演繹で導く、といった知識獲得の問題に帰着させている。

3512 タスクの特徴

設計型システムに分類される専門家の問題解決過程を情報処理技術の観点から眺めると、以下に示すような特徴を有している。

(1) 広い解空間

設計対象のシステム構造を決定することは、通常は組合せ問題となり、広い探索空間をいかに効率的に探索するか、探索空間をとう表現しておくかが課題となる。

(2) 多重の最適化

設計対象が複雑な構成をとる場合には、一般的には問題を階層的に分割しておく手法がとられる。このため、ある階層の構成要素を決定すること（最適化）は、さらに下位の階層の構成を設計する必要があるため、多重の最適化問題となる。これをいかにうまく階層分けし、全体として最適化していくかが課題となる。

(3) 検証の完全性

設計を階層的に分割する場合、上位階層と下位階層間での無駄な探索の繰り返しを避けるため、上位階層の構成条件が下位階層での必要条件を必ず満足するように構成しておく必要があるため、各階層内での検証の完全さが課題となる。

以下に、設計型システムの定義、問題の特徴、基本タスク、必要な問題解決機能を示す。

①定義

設計対象のシステムの入力と出力の要求仕様が与えられたとき、これを実現するために構成要素を組み合わせ、かつ各構成要素の内部仕様を決定すること。

②問題の特徴

1 システム構造の解空間が大
2 解析による合成が基本的
3 構成要素の最適化の必要性
4 検証の完全性
5 検証の効率性
6 対話形式での設計支援
7 設計段階に応じた支援形態

③基本タスク

1 構成要素とその関連の表現
2 設計問題の階層的表現
3 代替案の自動生成
4 部分システムの評価

5 上位レベルへの知的後戻り
6 設計事例の検索と再利用
7.並列的問題解決

④問題解決機能
1 トップダウン精密化戦略
2 拘束最小化戦略
3.最適化機能
4.検証機能
5.仮説推論による知的検索
6 類推による事例の利用
7.協調型推論

3 5.1 3 設計型システムのV&V

設計型システムには以上述べたような特徴があるため、次のような項目を重視した評価が不可欠である。

(1) 知識ベースの妥当性評価

設計型システムでは、専門家だけでなく一般の業務担当者も有する業務実施で参考となる常識的な背景知識は、量が多い、領域知識として意識していない、などから、エキスパートシステムにこれら全てを内蔵することが不可能となる場合が多いため、専門家の業務実施結果に比べ、無意味な結果の候補を生成する場合もある。また、設計対象のモデルをエキスパートシステム内部でどのように表現しておくかは、妥当性の高い結果が得られるかどうかに大きくかかわってくる、などから、知識ベースの妥当性評価は重要である。

なお、具体的な評価項目には、知識ベースの正当性、知識ベースの完備性、知識ベースの一貫性、知識表現の選択と適切さ、知識獲得方法の選択と適切さ、等がある。

(1-1) 知識ベースの正当性評価

(C01)知識源の確認

設計型システムでは、専門家の思考様式に忠実に従うと、探索空間が広いので、かなりの処理時間を要することが多い。これを解決するために、新規に対象問題の解法アルゴリズムを開発してしまう場合もある。後者を適用する際には、その検証法が前者と異なることもあるため、注意が必要である。

(1-2) 知識ベースの完備性評価

(*C1)問題解決の範囲

システムで取り扱い対象とする問題解決の範囲が明確であること。
より広い入力条件のもとでも結果の候補案を得ようとして、知識ベース化の範囲を広げ過ぎると、専門家の業務実施結果と比較し無意味な結果の候補を生成する場合もあるため、注意が必要である。

(*C2)深い知識の範囲

深い知識の範囲が明確であること。

専門家が有する全ての深い知識（原理・原則的知識）をシステムに持たせることは量が多い、領域知識として意識していない、などから現実的には実現できないので、注意が必要である。

(1-3) 知識ベースの一貫性評価

(C02)複数専門家の知識の整合性

問題解決の範囲を広げるために採用した複数の専門家の知識については、複数専門家のＡＮＤ知識なのか、ＯＲ知識なのか、などを明確にしておく必要がある。

(C07)ルールの一貫性のテスト

冗長なルールを残していないこと。

設計型システムでは、競合ルールや循環ルールを内蔵していると妥当な結果を得られないため、殆ど排除できるが、冗長ルールに関しては試験（テスト）段階では見つける工数をあまりかけないことも多い。しかし、後日新たなルールを追加したり、他のルールを修正した場合に冗長ルールか動作し、突然、異常な結果をもたらすことがあるため充分な注意が必要てある。

(1-4) 経験的知識を正当化する方法とその確認

(1-5) 知識表現の選択と適切さの確認

(C06)緻密さのレベル

設計対象のモデルを知識ベースてどのように表現しておくかは、設計結果に大きな影響を与える。しかしながら、現状の技術レヘルては限界があるため、知識を提供する専門家の熟練度と将来の拡張を含めた要求仕様との兼ね合いて、システム利用者と充分な調整をする必要がある。

(C07)知識表現の保守容易性

設計業務の担当者は業務の実施に伴い常に従来より効率の良い設計手法・手順を模索している。このため、専門家が自ら知識ヘースを改良・拡充できるような知識表現としておくことが必要てある。

(*C1)設計対象のモデル

設計対象のモデル、知識プリミティブは下記の観点から、一様てあり、また相互の整合性がとれていること。

- 設計対象・属性・値に関する粒度
- 階層構造の表現
- 装置構造のある部分・部品だけが細かくモデル化されていないこと。
  （ある部分・部品たけが大まかにモデル化されている）
- 階層構造が異なった観点てモデル化されていないこと。

(*C2)知識コンパイラ

知識コンパイラに誤りがないこと。

設計型システムては、専門家の思考様式に忠実に従うと、探索空間が広いので、かなりの処理時間を要する。このため、専門家の知識のモデルと、設計対象のモデルとをまとめて、他の表現形式（主に計算機の内部表現に近い形式）に変換（コンパイル）しておくことがあるか、この変換過程で

誤りが生しないように注意する必要がある。

(1-6) 知識獲得方法の選択と適切さの確認

(*C1)設計事例の参照

設計事例の参照が、問題解決タスクを左右しないこと。

知識獲得段階て設計事例を参照することは、設計対象のモデルや問題解決機能を短期間で明らかにできる利点がある。しかしながら、基本的な問題解決タスクを決定する際には、主たるタスクをとれにするかには充分注意することが必要である。例えば、設計型でのタスクの型を、新規に設計結果を生成する場合（新規生成）と、既存の設計結果を修正して結果の候補とする場合（事例修正）との、大きく２つに分けてとらえた場合、設計事例だけに着目すると前者が過去の設計事例の検索と再利用の側面を内在しているようにも解釈てきる可能性があるからてある。

(2) 問題解決・推論方式の妥当性評価

設計問題では、探索の空間が広いため、専門家の持つ問題解決過程を正確に推論手法に反映させないと要求仕様で規定された処理時間内ては結果か得られないことが多い。このため、問題解決・推論方式の妥当性評価は最も重要である。

特徴的な評価項目には、推論処理の妥当性、推論時間の妥当性、推論手法の選択と適切さ、学習機能の適切さ、等がある。

(2-1) 推論処理の妥当性評価

(*C1)専門家の思考様式

専門家の思考様式を反映した推論処理になっていること。

探索空間が広いため、専門家の持つ問題解決過程を正確に推論手法に反映させないと、結果が妥当であるか判定できない、要求仕様で規定された処理時間内では結果が得られないことが多い、などから注意が必要てある。

(2-2) 推論時間の妥当性評価

(*C1)探索範囲と推論時間

探索範囲と推論時間との関係が妥当であること。

運用テスト段階で、小規模の問題のみ試験確認すると、実運用ての大規模データ適用時に、システム利用者が要求する時間内に解を得られないことがあるので注意が必要である。設計対象に依存した探索範囲の大きさと推論時間との関係、出てくる結果の候補数、網羅探索の場合の探索空間の大きさの程度（例：10のベキ乗オーダ）などに注意する必要がある。

(*C2)推論処理と従来の手続き

推論処理と従来の手続きを主体とする部分との時間比率が想定されていること。

設計型システムでは、３次元図形処理、設計結果の製造可能性の確認（デザインルールチェック）、多量のデータベースなど、従来の手続きて記述てきる部分の処理時間も長いため、知識処理と手続き処理とを交互に処理移行を行うと、実用に耐えない場合もある。

(2-3) 推論手法の選択と適切さの確認

(C01)問題解決のモデルやパラダイムの適切性

凡化タスク(generic task)など基本的な推論手法を組み合わせてエキスパートシステムを実現することで、ある程度の問題解決能力を実現でき、現実的な解決策となる事も多い。

(*C1)推論手法の複雑さと処理時間

推論手法の複雑さと処理時間増大とのトレードオフを考慮してあること。網羅探索で探索空間の広さの目安は推定できる。これをもとに、要求条件の範囲内で現実的な推論手法を適用する必要がある。

(2-4) 不確実情報に対する処理技法の適切さの確認

(2-5) 学習機能の適切さの確認

(*C1)基本的な技術

設計型システムては、設計対象のモデルによっては基本的な原理・規則が比較的簡単で、現在でも実現可能な学習機能て充分実用になる場合もある。

(3) 結果の妥当性評価

設計型システムへの知識処理の適用動機として最適化に着目した場合が多いので、結果の最適性は重要となる。たたし、最適性を追求するあまり、専門家でも実施していない推論手法や評価項目を取り込むと、ある条件のもとでは、よりよい結果が得られるものの、専門家では簡単に許容できるような条件の変化にまったく対処できなくなる恐れがあるため、注意が必要てある。

特徴的な評価項目には、解の正解率、解の最適性、等の評価がある。

(3-1) 正解の評価

(C01)正解の定義

設計型ンステムでは、得られた解は、通常、制約条件を満たしているため、正解率を定義しないことも多い。但し、既存の結果を再利用（そのまま、あるいは修正）する場合は、正解率を、

〔正解率〕＝〔正解数〕／〔総試験（テスト）数〕（ある時点、トレント）

で定義し、所望の解が得られる程度・割合を評価することもある。

(C08)無意味な結果

設計型ンステムでは、探索空間が非常に広いため、専門家ても得られた結果が正しいと簡単には指摘できない場合がある。また、専門家が通常得たことがないが、制約条件を満たす解を出力する場合もある。後者の場合は、得られた解が妥当であるのか、制約条件が不足しているのか、ンステムの誤りなのかを確実に見極める必要がある。

(*C1)機能の推定

部品の合成結果から機能の推定ができること。

設計階層の違いによって、設計対象のモデルや設計結果の候補の表現が変化していくが、あくまでもその対象の集合は変化しないこと、などを利用する（設計対象のモデルの変化前後での設計対象の機能・動作状況の照合・確認）。

(*C2)専門家の学習効果

知識獲得段階の専門家の学習効果に注意すること。

知識獲得段階で専門家の知識の適用範囲を探ろうとして、非常に難しい条件のもとでの業務実施を想定したインタビューをおこなうと、専門家がこのインタビュー作業によって新たな学習をしてしまい、正解の定義を変化させてしまうこともあるので、注意が必要である。

(3-2) 解の網羅性評価

(3-3) 解の最適性評価

(C05)他の方法との比較

他のシステムまたはOR等の手法と比較が可能な場合は、どちらがより制約条件、評価基準を満足しているのかを検討すること。またこの他に、数値シミュレーション、ハードウェアシミュレーション、実装置の製造後の測定、各種評価ツールによる確認などがある。

(C06)局所的な最適解

探索空間が広いので、得られた結果が局所的な最適解になっている場合があり、条件をわずか変化させたり、初期条件を入れ替えただけで結果が異なる場合もある。

(C07)多目的な評価

設計型システムでは、最適性を評価する尺度を比較的明確に決めことができる部分があるため、システム化にあたっては特定の項目だけの最適性に着目する場合も多い。一方、最適性を追及するあまり専門家でも実施していない推論手法や評価関数の機構を取り込んでしまうと、ある特定の条件のもとでは、より良い結果が得られるものの、専門家では簡単に対処できるような条件の変化にシステムがまったく対処できなくなる恐れがあるため、注意が必要である。このため、評価の基準となる専門家の持つ評価尺度をどこまで計算機で実現したかを明確にしておくことも必要てある。

(*C1)制約緩和知識と解の最適性

制約緩和の知識や、組合せ的爆発の解消手段と解の最適性との関係が適切であること。

(*C2)結果の評価基準

結果の評価基準に関しては、人間では持てるが、計算機ては実現が難しい項目に依存していないこと。

〔例：配置の見栄え・美しさ、色のバランス〕

(3-4) 解の評価方法の適切さの確認

(4) 要求仕様に対する達成度評価

設計型システムを適用する業務では、設計結果の外観など、当該分野の一般の担当者が有しているような人間のもつパターン認識能力を利用した評価がまだシステム化できないため、開発にあたっては適宜要求仕様と現実的な解決策を利用者と調整しておく必要がある。また、ヒューマンインタフェースに関しては、設計結果を可視化するための要求の調整段階では、具体的なヒューマンインタフェースを作成し、利用者に確認しながら開発するのも一

つの方法である。
特徴的な評価項目には、要求仕様内容、等がある。

(4-1) 要求仕様内容に対する評価
(C06)実現可能性
設計では、計算機で実現できるタスクはまだ限定され、パターン認識などの人間が非常に得意な機能については、当面人間に作業させるのが現実的な解決策である。例えば、設計結果の外観など、該分野の一般の担当者でも有しているような常識的な背景知識の全て（日常的に身の回りで起こるような物理現象を経験することで分かるような大きさ、質・量、他との接続関係）をシステムに内蔵することはできない。しかし、新しい知識処理技術の適用により解決する機能もあるため、要求仕様をユーザと合意する際、当該機能を組み込むか否かに注意する必要がある。

(4-2) 要求仕様に対する達成度評価

(4-3) 要求仕様設定手段の確認

(5) 知識ベースの保守性評価
設計分野では、専門家は業務の実施に伴い、常により効率のよい設計手法そのものを模索・改良している。このため、専門家が自ら知識ベースを洗練化できるような使用・運用形態が理想的である。
特徴的な評価項目には、知識ベースの理解性、保守用ツールの整備状況、等がある。

(5-1) 変更箇所発見の容易性評価

(5-2) 知識ベースの理解性評価
(C02)モデルの素直さ
設計型システムでは、新たな設計仕様条件・項目の追加などについては運用を開始してから比較的短期間で要求が発生することが多いため、設計対象のモデルを分かりやすくしておく必要がある。
- 機能と構造との表現。
- 対象モデルの操作容易性（拡張、改版、追加、削除）。
- 対象モデルの変更範囲（可能性）の限界は明確になっていること。

(*C1)制約条件、評価関数
制約条件や評価関数が、分かりやすいこと。
- 制約条件や評価関数の複数モジュールへの分散配置はないこと。
- 制約条件や評価関数の変更容易性（拡張、改版、追加、削除）。

(5-3) 変更箇所の局所性評価

(5-4) 知識ベースの統一性評価

(5-5) 品質安定性の確認方法の確認

(5-6) 保守マニュアルの整備状況の確認

(5-7) 保守用ツールの整備状況の確認

(C01)ツールの整備

設計分野では、業務の実施に伴い従来より効率の良い設計手法・手順が明らかになる場合も多い。このため、専門家が自ら知識ベースを改良・拡充できるように保守用ツールを準備していくことが大切である。

(6) ヒューマンインタフェースの妥当性評価

設計分野では、計算機で実現可能な機能はまだ限定され、パターン認識などの人間が得意な機能については当面人間に作業させるのが現実的で妥当な解決策である。このため、人間と計算機との機能分担や操作が複雑となるとともに、操作者に理解しやすい図形・図面などを多用することも多いため、ヒューマンインタフェースは重要である。

特徴的な評価項目には、利用時のインタフェース、保守時インタフェースの良さ、等がある。

(6-1) 利用時のインタフェース評価

(C01)結果の表示（情報の内容に応じた表現形式<図形・図面>）

利用者に理解し易い図形・図面などを多用すると処理時間との兼ね合いで、実用的な応答時間が保証できなくなる場合もあり、注意が必要である。

(C02)入力の操作性

図形で表示された任意座標のポインティング手段など情報の内容に応じた入力形式に注意すること。

(6-2) 保守時のインタフェース評価

(C05)ビジュアル化

知識ベースの可視性を高める機能を持つこと。

設計対象のモデルを変更した場合は、設計対象を利用者に理解し易い図形、図面などを利用して可視化することは有効である。なお、設計対象が具体的な物体・物ではなく、抽象的な「機能」や、なんかの「概念」などの場合には、解説などを付加する必要もある。

(7) システムの総合評価

設計型システムでは、知識ベースと利用者インタフェースのみでシステムを構成することは非常にまれで、従来のソフトウェア開発手法に基づいて開発された計算機シミュレーション、データベース、ネットワーク、他のハードウェアなどと組合せることが多く、また、設計完了後の人手介在による誤りを避けるために、設計段階から製造段階までを広範囲にシステム化することが多いため、統合性を重視する必要がある。

特徴的な評価項目には、統合性、エキスパートシステムの成長可能性、ツール・言語選択の適切さ、等がある。

(7-1) テーマ選定の妥当性評価

(7-2) 利用者の満足度評価

(7-3) 統合性の評価

(C02)既存機能または既存システムとの連続性

従来のソフトウェア開発手法に基づいて開発された計算機シミュレーション、データベース、ネットワーク、他のハードウェアなどと連続性があること。

(7-4) 成長可能性の評価

(C05)拡張の予定

設計対象、業務の実施に伴い従来より効率の良い設計手法・手順を模索した結果としてのヒューリスティック、問題解決手法、など専門家が自ら知識ベースを改良・拡充できるように拡張を想定しておくことが大切である。

(7-5) 開発／保守用ドキュメントの評価

(7-6) 開発方法の適切さの確認

(7-7) ツール・言語選択の適切さの確認

(C03)高性能ツールの選択

初期のエキスパートシステム構築ツールはＬＩＳＰの処理系に依存して、従来ソフトウェアとの親和性が低かったが、第二世代のには上記機能を含めた統合性が向上している。また、開発初期にはエキスパートシステム構築ツールを使用するが、実用システム開発では統合性を重視して従来の手続き型言語で再記述することもある。

(*C1)コンパイル機能

コンパイル機能が充分であること。

処理時間の短縮をはかるため、知識ベース全体をエキスパートシステム構築ツールでバイナリコンパイルしたり、開発者がカスタマイズを行って、ある特定の知識表現専用のコンパイラを開発する場合もある。

(7-8) システム開発の生産性評価

(7-9) システムのコストパフォーマンス評価

(7-10)セキュリティ対策の評価

3 5 2 計画型システム

3 5 2.1 トメインの特徴

(1) 計画型システムの定義

計画型問題とは、与えられた目標を達成するために、利用可能な資源（人・金・物など）を時系列データとして組織化することてあり［Kobayashi 1990］、その問題を解決するシステムを計画型システムという。

(2) 計画型システムの実現方法

計画型問題の問題解決プロセスは、設計型システムと同様に、探索的側面、意思決定的側面、再利用的側面、知識獲得側面があり、これらを考慮して問題解決戦略を検討することになる。推論手法は生成検査法を基本とするが、この場合、状態の定義と状態遷移をうまく設定することがポイントになる。

3 5 2 2 タスクの特徴

(1) 問題クラスの同定

［Kobayashi 1990］では合成型問題について、対象のシステムを３クラスに分類している。クラス１はシステムの特性が与えられた時、サブシステムの特性およびそれを構成要素としたシステムの構造を求める問題てある。クラス２はシステムの特性とサブシステムの特性が与えられた時、それを構成要素とした最適なシステムの構造を求める問題てある。クラス３はシステムの特性とシステムの構造が与えられた時、最適なサブシステムの特性を求める問題てある。このなかで、計画型問題は一般にクラス２の問題として同定する。

(2) 問題解決戦略

計画型問題に対する基本的な問題解決アプローチは生成検査法に求めることができる。生成検査法とはシステム構造や構成要素特性を仮に生成し、システム特性の検査を行なうことを繰り返して、より良いシステムを探索していく方法をいう。

(3) 組合せ問題

計画型問題では、システム特性が与えられたとき、これを実現するシステム構造および構成要素特性は一般に無限に存在する。すなわち、組合せ爆発の問題があり、この問題をどう克服するかが本質的な課題となる。

(4) 最適化問題

計画型エキスパートシステムでは、数式モデルとして定式化が困難なものを扱うため、解析的に最適解を求めることはてきない。また、専門家自身もとのような計画が良いのかという評価尺度を厳密に定義することはてきないため、現実的な最適解はわからない。したがって、一般的に計画型システムては、拘束条件を満足する複数の可能解を効率的に導き、それらを比較することによって、採用解を求める方法をとる。

3.5 2 3 計画型システムのV&V

(1) 知識ベースの妥当性評価

知識ベースの妥当性評価では、システム化対象の固有のモデルて扱われる資源に対する基本的なアクティビティとなる、順序付・割当・分割・合成といった知識タイプから確認しなければならない。なお、順序付とは利用可能な資源の順序を与えること、割当とは資源を実際に割り当てること、分割とは要求を満たすように大きさのもつ資源を数個に分けること、合成とは要求する大きさに合致するように資源を組み合わせること、を意味する。

(1-1) 知識ベースの正当性評価

(C01)知識源の確認

原理・原則的知識については、計画担当者が利用する計画立案のマニュアルやトキュメント及び計画立案に関する法令・規則類から確認する。経験的知識は、計画者を観察したり、過去に立案した計画を分析することにより、知識を抽出することになる。このような知識は、計画者の試行錯誤の過程を正しく踏まえて抽出することが重要であり、立案途中て行っている計画者の意思決定事由を確認しなければならない。

(C03)知識の整理

計画型システムで取り扱う知識は、対象構造に関する知識、制約に関する知識、制御に関する知識、に分類され、これらの知識カテコリに照らし合わせて、知識を整理し、知識が正しく記述されているか確認する。

(1-2) 知識ベースの完備性評価

(C04)システムの取り扱う対象とする範囲の明確化

計画型問題は、探索をガイトする理論や知識が確立していないことが多く、取り扱う対象の知識を前もって完備しておくことは困難てある。明確に定義することができない知識に対してはは、システム利用者の協調介入によって取り入れることが可能てあることを確認する。

(1-3) 知識ベースの一貫性評価

(C01)外部環境の変化への対応

計画型システムは、技術的進歩や計画方針の変更といった環境の変化により、利用可能な資源や知識が異なるので、知識ベースの保守が頻繁に起こる。このため、知識の保守に対して、強いモデルが設定されていることを確認しなければならない。

(C12)整合性の確認

システム利用者の協調介入による知識ベースの変更か行われても、知識ベースの整合性が保たれる機能が揃っていることを確認する。

(1-4) 経験的知識を正当化する方法と確認

(C05)過去のデータからの検討

過去の計画実績が利用可能てある場合は、評価対象のシステムが専門家と同等の計画立案過程を辿るか確認することにより、知識の正当性の裏付けることがてきる。

(1-5) 知識表現の選択と適切さの確認

(C05)推論機構や外部モジュールによる処理の依存度

システムの問題解決能力が、知識の増加によって向上しなくなることがあることに注意しなければならない。この原因には、知識相互の矛盾による誤答、冗長な知識の増加による実行速度の低下、推論制御に関する知識の増加による全体動作の不透明化、等がある。これらは高機能な推論機構によってある程度軽減することができるが、事前に対象領域の知識全体を整理・統合して、適切な知識表現を選択することが必要である。

(1-6) 知識獲得方法の選択と適切さの確認

(C01)専門家の選定方法の妥当性評価

計画業務の経験年数や業務遂行能力等を考慮して、知識を提供する専門家を選定する。知識提供者が成長段階であるときは、インタビューする毎に計画立案の考え方が変わることがあり、知識の整理・体系化が難しくなるので注意しなければならない。

(2) 問題解決・推論方式の妥当性評価

計画型システムの推論メカニズムに対しては、効率的に組合せ問題を解くタスクが実現されていること、不適格な解候補が生成されないこと、を確認することによって妥当性を検査する。

(2-1) 推論処理の妥当性評価

(C06)推論過程のトレース

解空間が非常に広く、計画パラメータの選択に自由度があるため、探索時に効果的な枝刈りが出来ることが重要となる。このため、推論処理の動作が透明であること、探索の刈り込みを行ったときの事由が適切であることを確認するためのトレース機能が必要である。

(2-2) 推論時間の妥当性評価

(C01)専門家のイメージ

診断型システムや制御型システムに比べて、推論時間に対してはシビアではないが、推論処理は実時間で行われることが要求される。

(M04)生成検査法の場合

生成検査法による推論時間の見積りは、次式で与えられる。

推論時間 ＝ 生成される探索ノード数 × 単位時間の処理時間

(2-3) 推論手法の適切さの確認

(C01)問題解決のモデルやパラダイムの適切性

計画パラメータによって、生成しうる可能解の数を把握し、パラメータの増大が実時間で解くことができなくなる可能性がある場合には、効率的な探索戦略を採用する必要がある。探索を効率化するための戦略として、階層的生成検査法、トップダウン精密化、拘束最小法、関係依存後戻り、がある。

(2-4) 不確実情報に対する処理技法の適切さの確認

(C01)結果の妥当性・保守性の低下

実用システムを構築する場合は、結果の保証や保守性の観点からみると、一般に不確実情報を用いることは得策ではない。しかし、計画型システムでは予測や判断を行う際に、入力データ及び利用する知識が特定できない場合は不確実な情報処理が必要である。このとき、結果の保証の問題については、利用者の理解が得られていることを確認しておかなければならない。

(2-5) 学習機能の適切さの確認

(*C1)基本的な技術

計画型システムでは、問題解決の再利用的側面からの技術的アプローチとして事例ベース推論がある。事例ベース推論を実現するには、現在の問題の特徴の理解とその特徴に基づいて過去の類似事例を検索し、その解を現在の計画立案に利用できるように変更することが可能てなければならない。

(3) 結果の妥当性評価

計画型エキスパートシステムは数式モデルとして定式化が困難なものを対象とするため、一般には解析的に最適解を求めることはできない。このため、結果の妥当性では、専門家と同等の解が得られることを確認する必要がある。要求される解は最適解てはなく、実行可能解の場合もある。

(3-1) 正解の評価

(C01)正解に関するコンセンサス

計画型システムでは、すべての制約条件を満足する計画案が実行可能解とならなければならない。つまり、与えられた資源の割当がすへて完了した時に、制約条件を満たしているものが正解として良いのか確認しておかなければならない。

(3-2) 解の網羅性評価

(C01)網羅性の目標

計画型ノステムては通常、組合せ爆発の問題があり、この問題をどう克服するかが本質的な課題となる。効率的な探索を行う際に、採用解の可能性がある候補案を棄却することがないよう注意しなければならない。

(3-3) 解の最適性評価

(C07)多目的な評価

例えば、生産スケジュールでは、コスト減少と納期日遵守、待ち時間最小化と機械稼動率増大、等がトレードオフの関係になる。どちらか一方が優先する場合は問題ではないが、通常は双方の妥協点を探りながらスケジューリングを行うため、状況に依存して設定される評価関数が妥当であるか確認しなければならない。

(3-4) 解の評価方法の適切さの確認

(C01)評価方法

過去の計画事例がある場合には、それを利用して確認する。できるならば、人間による解の確認、統計やＯＲの手法を用いた確認、等も検討し、導出される解が妥当であることを示す評価方法を用意しておく。

(4) 要求仕様に対する達成度評価

計画型問題は、通常、悪構造(Ill-structured)であり、システム化が容易ではない。つまり、問題の構成要素が多種・多様である、要素内の関連が複雑である、制約条件が動的に変化し複雑である、等により、モデル化することが困難である。このため、プロトタイピングアプローチを用いることによって、計画立案過程の分析、計画立案戦略の定式化、対象モデルの記述、仮説集合の同定、等の要求定義に必要な項目を明確にしていくことになる。

(4-1) 要求仕様内容に対する評価

(C01)正当性

システム開発のライフサイクルの各フェーズで要求されるシステムの規模が妥当であることを確認し、たとえシステム規模が大きくなっても、組合せの爆発が生しない手段を用いていることを検査する。

(C02)完全性

システムの利用者が誰であるか、明確であることを確認する。利用者が非専門家の場合は、システム上に知識をすべて記述しておかなければならない。利用者が専門家の場合は、核となる計画案は専門家が入力し、システムはそのイメージを保証するように未決定部分の補完を行って計画案を生成することになる。

対象とする問題の解空間の広さも確認する必要がある。解空間が狭い問題（列車ダイヤ編成、フライトスケジュール等）は、制約条件を緩和していくことにより、実行可能解を求めることになる。解空間が広い問題（生産スケジュール等）は、決定変数に対して制約条件がそれほど厳しくなく、実行可能解が無限にある。この場合は探索空間の効率的な刈り込みが必要になる。

(4-2) 要求仕様に対する達成度評価

(C01)～(C05)各フェーズにおける要求仕様に対する達成度

システム開発のライフサイクルの各フェースにおいて、システムへの要求事項を達成しなければならない。例えば、プロトタイプのフェースでは、効率的な探索と最適化を目的とした計画立案で用いる知識の実現が主な達成項目であり、実用システムのフェーズでは、さらに、操作性、応答性、保守性、等が加わる。

(4-3) 要求仕様設定手段の評価

(C02)プロトタイプの利用

システム開発のライフサイクルに応したプロトタイプシステムを構築することによって、悪定義の部分がどれだけ見通しが良くなったか確認し、それが次のフェーズの要求仕様の設定にとのように生かされたか検査する。

(5) 知識ベースの保守性評価

計画型システムは、外的環境の変化や計画方針の変更により、利用される知識が異なったり、陳腐化したりするため、知識ベースの保守か頻繁になる。したがって、知識ベースの保守性評価は重要な項目である。

(5-1) 変更箇所発見の容易性評価

(C04)トレース機能

変更箇所を容易に発見するためには、知識ベースに格納された各々の知識及び知識間の関連が容易に理解できること、変更による複雑な属性間への影響の伝播が把握できること、が必要である。このため、推論過程をトレースする機能は不可欠である。

(5-2) 知識ベースの理解性評価

(C01)専門家による保守

多くの計画型システムは事前に知識を完備させておくことは困難であるため、計画者が知識を追加、変更しながら、システムと協調的に計画を立案していくことになる。このため、なるへく物理的対象に即したモデルを採用し、プログラムに関する知識がない計画者ても直接、知識ベースを操作できるようにしておくことが望まれる。

(5-3) 変更箇所の局所性評価

(C06)変更可能な範囲

知識を変更することによって、組合せの爆発が生したり、無意味な解を導出したりすることがないようにするため、保守可能な知識の範囲を明確にしてあることを確認する。

(5-4) 知識ベースの統一性評価

(C01)対象のモデルの明確性

(1-2)参照。

(5-5) 品質安定性の確認方法の評価

(C04)変更による影響伝搬

(5-1)及び(5-3)参照。

(5-6) 保守マニュアルの整備の状況の確認

(C01)保守マニュアル

(7-5)参照。

(5-7) 保守用ツールの整備状況の確認

(C01)ツールの整備

(5-1)及び(5-2)参照。

(6) ヒューマンインターフェースの妥当性評価

ヒューマンインターフェースについては、システムが高レヘルの応答能力があるかどうかが評価される必要がある。なぜなら、多くの計画型システムのタスク

では人とンステムが協調的に作業を進めることを要求されるからてある。

(6-1) 利用時のインターフェース評価

(C01)結果の表示

計画型システムては、利用者が容易に介入てきる環境が整っていることが必要である。つまり、立案中の計画がわかりやすく画面上に表示され、利用者の注意が、その計画の改良すへき点に集中てきること、さらに修正操作がマウス等て簡単かつ迅速に行えること、を確認する。

(6-2) 保守時のインターフェース評価

(C02)ビジュアル化

知識ベースの保守性評価の各項目を考慮し、保守時において、理解性、操作性の良いインターフェースの環境が整っていることを確認する。

(7) システムの総合評価

知識処理システムの多くは計画型システム全体の一部分に過ぎないため、他ンステムとの統合性など、対象とするシステム全体の総合評価が重要になる。

(7-1) テーマ選定の妥当性評価

(C01)ニーズの強さ

まず知識処理の必要性があるほど悪構造な問題であるか検討しなければならない。悪構造な問題とは、問題の構成要素が多種・多様てある、要素内の関連が複雑である、制約条件が動的に変化し複雑である、評価関数が厳密に定義できない、といったことてある。さらに、数式モデルとして解析的に解ける問題を扱っていないことを確認しなければならない。たたし整数線形計画法のように、モデル化が可能であっても実時間て解けない問題はエキスパートンステム化の対象になる。

(7-2) 利用者の満足度評価

(C03)導入効果、利点

このシステムを利用することによって、従来システムと比へ、迅速化、省力化が実現され、最適性を重視した計画の立案に集中てきるようになったか確認する。

(7-3) 統合性の評価

(C02)既存機能または既存システムとの連続性

知識処理システムは、他の従来システムと統合していることが多く、他システムとの連続性を確認する必要がある。具体的には、他のンステムて実現されている統計処理、ＯＲ等の計算処理をサブタスクとして使えること、推論処理に必要なデータを他ンステムから取り込めること、等がある。

(7-4) 成長可能性の評価

(C01)レビュー

ンステムの対象領域範囲内において生産設備の大幅な更新やトラスティックな計画方針の変更といった環境変化が生じた場合に、どの程度対応がで

きれば良いのか、開発チームのメンバー間で、同意が得られていることを確認する。

(7-5) 開発／保守用トキュメント評価

(C01)必要なトキュメント

システムの開発・運用に携わる人、各々に必要なトキュメントが揃っていることを確認する。例えば、利用者には外部仕様書であるマニュアルが必要である。開発者には、基本仕様、機能仕様、構造仕様、等を正確に記述した内部仕様と開発状況を記録した開発管理書が必要である。運用者には知識システム内部の構造、論理の理解を助けるための内部構造説明書や問題分析説明書が必要である。

(7-6) 開発手法の適切さの確認

(C04)問題に適した開発方法

(4-1)及び(7-1)参照。

(7-7) ツール・言語の適切さの確認

(C02)一貫した開発

(1-5)及び(4-2)参照

(7-8) システム開発の生産性評価

(C01)生産性指標

計画型システムは、開発初期では必要な知識の総量が明確ではなく、知識ベースの規模の推定を行うことは難しい。さらに、システムが成長するとともに知識ベースが拡張することを考慮すると、コスト、時間等のシステム開発の見積りは充分に検討して設定しなければならない。

(7-9) システムのコストパフォーマンス評価

(C02)作業の質

システム化によって、計画立案に関わる時間と計画の質がとれたけ改善されたか把握し、それが投資に見合ったものであるか確認する。

(7-10) セキュリティ対策の評価

(C01)充分な対策

知識ベースは計画者のノウハウの集積であるため、その機密保持の対策がなされているか確認する必要がある。

3.5.3 解釈型システム

3.5.3 1 ドメインの特徴

(1) 解釈型システムの定義

解釈型問題とは、計測器あるいはセンサを通して観測されたデータを分析してシステムの状態を推定し、これに物理的な意味づけを与えることであり[Kobayashi 1990]、その問題を解決するシステムを解釈型システムという。

(2) 解釈型システムの実現手法

解釈型システムの実現に当たって考慮すべきことがらを、解釈型システムの特徴、基本タスク、問題解決機能の３つの観点からまとめる[Kobayashi 1990]。

(2-1) 解釈型システムの特徴

①時空間データの高次相関性

大規模・複雑な対象では、計測データが多次元に及び、それらが時間的にも高次の相関を有するために、従来の信号処理の結果たけからは、物理的に意味のある帰結を得ることが困難である。

②不規則な雑音の混入

計測データの中に、従来のフィルタリング技術だけては除去てきない不規則な雑音が存在する場合、そのような雑音の発生源および原因を追求することにより、これを処理しなければならない。

③誤りの存在

データ中に誤りが存在する可能性が常にある。したがって、誤りが存在することを前提として推論を行い、もし矛盾が生じた場合には後戻りして、誤りの箇所を同定して、これを修正しなければならない。

④必要なデータの欠落

解釈に必要なデータが完全にそろっているのはむしろ例外的なことてあり、必要なデータが一部欠落していることが普通てある。そのような状況のもとても解釈が行えるような頑健性が要求される。

⑤信号処理技術との接続

計測器から取得されたデータが直ちに知識処理の対象とされるのは例外てあり、通常はデータの前処理において従来の信号処理技術の利用が必要てある。したがって、信号処理技術と知識処理技術の間での役割分担およひその接続が要請される。

⑥対象に関するモデルの利用

解釈型知識システムては、対象の理解に基づいての処理、すなわち対象のシステム構造、機能および挙動に関するモデルを構築して、これを陽に利用して問題解決を行うことが基本てある。

⑦感覚データの取扱い

視覚、聴覚、臭覚など人間が得意とする感覚データに基づく判断が大きなウェートを占めている問題では、これを知識処理だけで解決することは極めて困難であり、いくつかの研究課題を残している。

(2-2) 解釈型システムの基本タスク

(2-1)にまとめた特徴を持つ問題をシステム化するための基本的なタスクは、以下の通りである。

- 特徴の抽出
- モデルとの不完全な照合
- システム構造の同定
- システム状態の推定
- あいまいさの処理
- 不完全性の処理
- 解釈の多様性

(2-3) 解釈型システムの問題解決機能

(2-1)にまとめた特徴のシステムが有する問題解決機能は、以下の通りである。

- 分類階層によるクラス分け
- モデル検索機能
- 部分構造の評価機能
- 階層的生成検査法
- 不確実性推論
- 仮説推論
- 協調型推論

3 5 3 2 タスクの特徴

ここでは解釈型システムの評価を念頭に置いて、そのタスクの特徴を考えてみる。

解釈型のシステムは診断型、制御型と並んで広い意味で解析型と呼ばれることが多いが、形態としては単独で存在する場合よりは他のシステムと協調して機能することが多い。実際、ほとんどの場合、解釈型のシステムは診断型、制御型のシステムの前処理的なタスクを受け持つと考えられる。診断型システムでは異常の有無とその対策を推論するために診断対象からのデータを基に対象の現在の状態を推定しなければならない。また、制御型システムにおいても制御対象の状態を推定することは重要な要素である。換言すると、診断型、制御型のシステムはその中に解釈型システムのタスクを包含することが一般的と言える。したがって、解釈型システムの特質を考えるためには他の解析型システムの特質を併せ考えなければならない。

以下に解釈型システムの評価に当たって特に考慮すべき、タスクの特徴を整理する。

(1) 実時間性

解釈型システムは診断型、制御型システムの前処理的なタスクを実行するこ

とが多く、しかも後続の処理がタイムクリティカルな性格を持つことが一般的であるため、実時間性が重要である。

(2) モデルと現実のシステムとの親和性

解釈型システムは一般的にはあらかじめ与えられたモデルに基づいて作成された有限の解空間の中から、対象システムの状態を推論することが多い。解空間が非常に広いかまたは無限であるような設計型問題とはこの点が異なる。したがってあらかじめ与えられるモデルと現実のシステムとの親和性が解釈型システムを正確かつスピーディに実行するための必要条件となる。

(3) 組み込み型システム

診断や制御までを含んだ組み込み型システムは技術的な難易度が高く、現時点では現実的とは言えないが、解釈型システムとしての観点で見ると組み込み型システムの要請は高く、また将来的には診断や制御までも含んだ形での組み込み型システムが中心的な課題になると考えられる。

(4) 高品質

解釈型システムが誤りを犯すと、後続することの多い診断、制御などのシステムの処理がすべて意味をなさなくなってしまう。致命度という観点で考えると解釈型システムの誤りが他のシステムに対して決定的な重要性を持っている。したがって、高品質はもちろんのこと、成熟性、回復性なども重要である。

(5) 大量の時系列データ

センサなどを介して得られるデータは時系列的で大量であることが一般的である。こうしたデータを素早く処理して対象システムの状態を推論することが必要である。

(6) 頑健性

解釈型システムの直接的な入力となるデータはセンサなどから得られ、大量でしかも誤りを含むことが多い。したがって、解釈型システムの推論機構は誤りデータに対して頑健性を持っていなければならない。

3.5.3.3 解釈型システムのV&V

3.5.3.2に整理した特徴を勘案して、解釈型システムで特に考慮すべき評価項目などについて以下にまとめる。

(1) 知識ベースの妥当性評価

モデルと現実のシステムとの親和性、高品質を保証するという観点で知識ベースの妥当性を評価する必要がある。

(1-1) 知識ベースの正当性評価

(C01)知識源の確認

解釈型システムは診断・制御などのシステムあるいは信号処理などの従来技術によるシステムなどと複雑に関連していることが多い。したがって、

異なった観点からの知識を統合する必要が生ずることがあると考えられる。

(1-2) 知識ベースの完備性評価

(*C1)テスト項目の欠落チェック

解釈のタスクの品質を保証するという意味てテスト項目の欠落をチェックする必要がある。

(C04)システムの取り扱う対象とする範囲の明確化

関連または後続する他のタスクとの関連でンステムの対象範囲を明確にしておく必要がある。またンステムが完成に近つくにつれて対象範囲が変化することが一般的であることも考慮する必要がある。

(1-3) 知識ベースの一貫性評価

(C01)外部環境の変化への対応

解釈型システムが扱う対象の機器構成の変化なとによってモデルが変化することが多い。こうした事態に対応できるような知識ベース構造が必要てある。また、解釈型システムが異なった場所やハートウェアて運用されている場合などにも知識ベースの同一性が保たれていることを保証てきなければならない。

(C02)対象モデルの一貫性

知識の記述の前提となる対象モデルが時間の経過、関与者の相違のためにまちまちになっていてはならない。

(1-4) 経験的知識を正当化する方法とその確認

(*C1)経験的知識の適用範囲

経験的知識の適用範囲を過度に考え、解釈型システムの対象の構成や機能の変化に対して一般化させすぎてはならない。

(1-5) 知識表現の選択と適切さの確認

(C01)知識表現手法と対象問題の特徴の関連

対象システムを適切にモデル化することが可能な知識表現手法を選択する必要がある。

(C04)応用分野の特性を考慮した妥当な知識表現の採用

対象の正しいモデル化、推論処理の速度向上のためには応用分野の特性を十分に考慮した知識表現が必要てある。特に、信号処理やパターン情報処理などとの関連を考慮する必要があることが多い。

(C05)推論機構や外部モジュールによる処理の依存度検討

大量データの処理やパターン情報の処理などには外部のモジュールの処理が必要になることが多い。

(1-6) 知識獲得方法の選択と適切さの確認

(C01)専門家の選定方法の妥当性評価

解釈型システムは診断・制御などのシステムあるいは信号処理なとの従来技術によるシステムなどと複雑に関連していることが多い。したがって、異なった観点からの知識を統合する必要が生ずることがあると考えられる。

(*C1)不足している知識の領域

関連システムとの整合性を保つ上から、カバーてきていない知識の領域を明確にしておく必要がある。

(2) 問題解決・推論方式の妥当性評価

実時間性を達成するために高速な問題解決・推論方式か要求される。高品質を実現するためには誤りの無い、正確な推論を行う必要がある。

(2-1) 推論処理の妥当性評価

(*C1)頑健性

誤ったデータに対しても一定の動作が保証されるような、フェイルセーフ的な推論機構になっている必要がある。

(2-2) 推論時間の妥当性評価

(*C1)総合的な推論時間の妥当性

診断や制御などの関連システムと合わせて総合的な観点からの推論時間の妥当性を考える必要がある。

(M03)問題の規模増加に対する推論時間の増加オーダー

対象システムの構成変更などに対して非線形に推論時間が増大するような事態を避けなければならない。

(2-3) 推論手法の選択と適切さの確認

(C01)問題解決のモデルやパラダイムの適切性

推論の正確性、迅速性、頑健性などを保証するために最適な推論手法を考慮しなければならない。

(*C1)関連するアルゴリズム的な処理の使用

信号処理などのアルゴリズム的な処理が要求されることが多いので、こうした従来技術によるシステムを統合できるような仕組みが必要である。

(2-4) 不確実情報に対する処理技法の適切さの確認

(*C1)不確実情報を扱うことの意義

現状の技術では不確実な情報を適切に扱うことは難しいため、不確実な情報を扱うことの意義を明確にしておくことが望ましい。たとえば、可能性のある解を多数出すことによって、間接的にシステムの品質を高めるというような場合が有り得る。

(2-5) 学習機能の適切さの確認

(C05)学習機能による処理時間の短縮

高速な推論処理を実現するために学習機能が有効てあることがある。

(3) 結果の妥当性評価

解釈型システムの致命度や実時間性を考えると結果の妥当性の評価が必要てある。

(3-1) 正解の評価

(*C1)解の安全側評価

後続のタスクとの関連て解は安全側に評価しておく必要がある。たとえば、解釈型のンステムで見落とした異常状態は、診断や制御なとのタスクではまったく考慮されないことになる。

(M01)正解率=正解数/総テスト問題数

(3-2) 解の網羅性評価

(C04)解の順番性

解釈型システムは全体ンステムの入り口に当たる処理を行うことが多いので、後続タスクの処理の効率を考えて、解候補を良い順番に出力することが望ましい。

(3-3) 解の最適性評価

(*C1)診断・制御なとのタスクの特性との整合性

解の最適性は解釈型ンステムのみて考えるのてはなく、関連のタスクも含めた全体ンステムて考える必要がある。

(3-4) 解の評価方法の適切さの確認

(4) 要求仕様に対する達成度評価

解釈型ンステム、あるいはそれを含んだ診断型システムや制御型ンステムの要求仕様が初めから完全な形て与えられることは稀て、プロトタイプンステムなとによって潜在化していた診断対象や制御すへきケースが明らかになることが多い。したがって、固定された仕様に対する達成度を考えるよりは、システムの柔軟さ、変更に対するロバストネスを考慮しておくへきてあろう。

(4-1) 要求仕様内容に対する評価

(4-2) 要求仕様に対する達成度評価

(4-3) 要求仕様設定手段の確認

(5) 知識ベースの保守性評価

(4) で述へたンステムの柔軟さ、ロバストネスを保証するために保守性を評価しなければならない。

(5-1) 変更箇所発見の容易性評価

(5-2) 知識ヘースの理解性評価

(5-3) 変更箇所の局所性評価

(5-4) 知識ベースの統一性評価

(*C1)診断・制御などのタスクの知識ベースとの整合性

解釈型システムに関連する診断・制御などのタスクの知識ベースも同時に保守する必要があるため、これらの知識ベース間で知識の整合性が保たれていることが望ましい。

(5-5) 品質安定性の確認方法の確認

(C01)標準問題集の有無

入力データとして大量の時系列データなどが要求されるため、標準の問題集を作成しておくことが望ましい。

(C02)動作確認用のサンプルデータの有無

動作確認には多くのバリエーションを持った入力データが必要である。特に、特定のタイミングに依存するデータや誤りを含んだデータも動作確認には不可欠である。

(5-6) 保守マニュアルの整備状況の確認

(5-7) 保守用ツールの整備状況の確認

(6) ヒューマンインタフェースの妥当性評価

解釈型システムは組み込み型であることが特徴のひとつであることからも想像できるように、概して人間に対するインタフェースは処理に実時間性が要求される以外はそれほど必要としない。

(6-1) 利用時のインタフェース評価

(*C1)取扱い範囲の限界を認識し適切に応答

対象システムの機器構成の変化や予期せぬエラーデータの混入などによって、知識ベースの対象範囲を逸脱した場合には、それを認識し適切に応答できることが望ましい。

(6-2) 保守時のインタフェース評価

(7) システムの総合評価

解釈型システムの特質は実時間性、モデルと現実のシステムとの親和性、組み込み型システム、高品質、頑健性である。個々のシステムはそれぞれ場合によって目標の重点が異なると考えられる。こうした状況を考慮して、実時間性と高品質、組み込み型と柔軟性などの相反する特性をコントロールして、場合に応じて最適な特性を備えたシステム構築を行うことが、総合システムとしての完成度につながる。

(7-1) テーマ選定の妥当性評価

(7-2) 利用者の満足度評価

(C11)出力結果の正当性

設計型システムなどと異なり、解釈型システムでは結果の正当性に対する

要請が極めて強い。

(7-3) 統合性の評価

(C02)既存機能または既存システムとの連続性

信号処理などの既存システムと密接に連動する必要のあることが多く、それによる利点を引き出すことが求められる。

(7-4) 成長可能性の評価

(*C1)適用対象分野の新プロセス、新テクノロジへの対応の柔軟性

適用対象システムの変化に対応し易い知識ベースになっていることが必要である。

(7-5) 開発／保守用ドキュメントの評価

(7-6) 開発方法の適切さの確認

(*C1)開発時の環境

解釈型システムは入力データとして大量の時系列データ、連動するシステムとして信号処理などの既存システムや診断・制御型エキスパートシステムが必要になることが多い。したがって、開発時にもある程度こうした環境が整っていることが要求されることがある。

(7-7) ツール・言語選択の適切さの確認

(7-8) システム開発の生産性評価

(7-9) システムのコストパフォーマンス評価

(7-10)セキュリティ対策の評価

3 5 4 診断型システム

3.5 4 1 トメインの特徴

(1) 診断型問題の定義

診断型問題とは、対象とするシステムに異常または故障が生した場合に、観測されたデータおよび対象システムに関する知識を利用して原因の箇所を同定し、正常な状態への復旧をはかることであり［Kobayashi 1990］、その問題を解釈するシステムを診断型システムという。

ここでは、主にエンジニアリングシステムを対象とした診断型システムの評価について考える。

(2) 診断型システムの特徴

診断型システムは、以下の点で特徴付けられる［Kobayashi 1990］。

①設計レベルの知識の利用

プラントや機械設備などのエンジニアリングシステムでは、対象の構造に関する「深い知識」を利用することができる。深い知識を導入することにより、診断の完全性、未知の状況への対応、適用範囲の判断、などの機能の実現が期待できる。ただし、深い知識を利用するために定性推論などの手法を用いる場合には、推論方法の妥当性に注意が必要となる。

②経験的知識の利用

専門家は、故障原因に特有の兆候や故障の頻度など診断についての経験的な知識を、診断作業の効率化などに利用している。特に、対象システムのふるまいが解析できない場合には表層的な因果関係の知識に依存せざるを得ない。このように診断型システムにとって経験的知識は重要な位置つけにある。しかし、経験的知識について正当性、完備性、一貫性といった妥当性を評価することは経験的であるがゆえに困難な場合が多い。知識ベースの妥当性評価についての対処を、開発の早い時期に定めておく必要があろう。

③経験的知識の不確実性

経験的知識は不確実性をともなう。また、観測データにも不確実性か含まれることが想定される。これらの不確実性を推論の側て適切に取り扱うことが必要となる。知識ベースの規模が大きくなると、局所的に定めた不確実性の指標が最終的な結論にどのように影響するかを見通すことが困難になり、保守性の低下を招く恐れがあるため注意が必要である。

④操業データの利用

プラントや機械設備などの故障診断では、操業データを故障発生時の初期診断データとして利用てきる。また、過去の故障履歴から故障の傾向に関する知識を得たり、モニタリングによって故障の予知に利用てきる。

⑤計測のための時間とコスト

故障原因の同定のためには対象システムのふるまいを計測する必要がある。

計測には時間とコストがかかる。このため、計測によって得られる情報と、時間およびコストとのトレードオフを要求仕様の段階から考慮しておく必要がある。なお、作業性から計測の手順が制限される場合もある。

⑥知識の抽象化の必要性

診断対象システムの構造と機能に関する知識は様々な抽象レベルでとらえることができる。診断問題では、対象システムにとって例外的な状況を扱うため、全ての場合を網羅することは不可能である。対象システムの規模および診断結果に要求されている詳細さの度合と、知識の量および推論に要する計算量などを考慮して適切なレベルで抽象化する事が必要である。

このため、診断システムの目的に応じて対象システムを適切な抽象化レベルでとらえる必要がある。

⑦対話的診断の推論制御

診断型システムの目的は、正常状態への復旧をはかることである。現状では、実際の復旧作業は専門家あるいはオペレータなどによって行われ、エキスパートシステムはそれを支援する役割を担っている場合が多い。このため、利用者とシステムとの対話性が実用性を大きく左右する。利用範囲、経験レベル、といったユーザモデルを明確に定めて、それに合わせた適切な推論制御を行う必要がある。特に、説明機能、動作の透明性、冗長な質問の抑制、利用者主導の対話、応答性の確保、などが重要である。

3 5 4 2 タスクの特徴

①異常事象の分類階層的表現

対象システムの状態を推定するため、異常事象を分類階層木として表現するタスクである。適切に階層化することにより、診断の見通しが良くなり、また、診断システムの保守性も向上する。なお異常事象の分類階層的表現と、診断実行時にシステムがどのように質問をたしていくかという質問制御戦略とは、区別して考えることが必要である。

②システム構造の階層的表現

対象システムの構造に関する知識を、物理的あるいは機能的な観点から階層的に表現するタスクである。この表現は、診断推論に直接利用される場合と、診断知識の抽出に利用される場合とがある。

③解釈問題のタスクを内包

測定データを診断推論の入力として利用可能可能な形式にする解釈のタスクである。データの特徴抽出、モデルとの照合、システム状態の推定、あいまいさの処理、不完全性の処理、最適な解釈の選択、などのタスクが必要となる。

④計測点の選択的決定

故障原因を追求していく過程で効率的な計測点を順次選択的に決定するタスクである。対象システムの構造に関する知識を利用し、計測から得られる情報と計測に要するコストとのバランスを考慮して決定する。

⑤異常原因の同定

システムの構造と構成要素の特性とをふまえて、測定データから異常原因を同定するタスクである。診断推論の非単調性や、診断知識の不確実性などに留意する必要がある。

⑥浅いモデルによる効率性

経験的な知識を利用して異常原因の近傍へすばやくアクセスするタスクである。対象システムが大規模な場合や、応答速度が重視される場合に重要となる。また、診断システムを現場へスムーズに導入するためにも専門家にとってなじみのある浅いモデルの利用は有効と考えられる。

⑦深いモデルによる完全性

対象モデルの設計レベルの知識を用いて、ある抽象化レベルての完全な診断を行うタスクである。測定データを最も合理的に説明する仮説を生成する考え方などが提案されている。

3.5 4 3 診断型システムのV & V

以上に述べた特徴から、診断型システムのV & Vでは3 1で述べた評価項目の中で次のような点がポイントとなる。

(1) 知識ベースの妥当性評価

診断型システムには、異常な状態から正常な状態への復旧を支援するために、多岐にわたる診断の知識が蓄積されている。システムが実用化されるたけの信頼を得るためには推論結果の妥当性だけではなく、知識ベースがとの様に作られているかというところまでさかのぼって妥当性を評価する必要がある。知識ベースの正当性はもちろん、経験的な知識の正当性や知識獲得への系統的な取り組みなどにも注意が必要である。

(1-1) 知識ベースの正当性評価

(C01)知識源の確認

知識の出所（修理マニュアル、オーソリティなと）を参照できるようにしておくことは診断型システムの正当性にとって特に重要である。

(C07)専門家間での合意

複数の専門家から知識獲得を行い合意を得るプロセスは知識の正当性確保に有効である。

(C10)結果の視覚化

システムの振る舞いを視覚化して専門家による確認を容易にすること。

(*C1)設計レベルの知識を利用

エンジニアリングシステムを対象とした診断型システムでは、診断知識の正当性を設計レベルの知識を利用して可能な限り確かめる。設計レヘルの深い知識から浅い知識を生成する知識コンパイラの研究もなされている。

(*C2)FTAやFMEAなどによる解析との対比

システムの安全解析に用いられているFTA(Fault Tree Analysis)やFMEA(Failure Mode and Effects Analysis)などの手法による結果と対比する。

(1-2) 知識ベースの完備性評価

(C01)知識の分解レベル

診断型システムを利用する環境に応じて、観測できる兆候や修理するモジュールの単位などが定まる。診断知識ベースはこれらの前提条件をふまえた上で完備性を備えている必要がある。

(C04)システムの取り扱う対象とする範囲の明確化

診断型問題では対象システムにとって例外的な事象を取り扱うために、知識の完備性を求めることは困難である。診断の実務に即して重点を定めて段階的に整備して行くことが現実的と考えられる。

(C07)完備性のテスト

重要な属性の欠如などのルールの不備を過去の測定データによるテストなどにより除去すること。

(1-3) 知識ベースの一貫性評価

(C08)ルールの条件部の適切さ

ルールの条件部を実験結果から自動的に一般化あるいは特殊化するような支援システムの利用により効率的な知識ベースの洗練が可能である。

(1-4) 経験的知識を正当化する方法とその確認

(C02)正当化理由

診断対象システムの機能構造モデルに基づく深い知識と、経験的知識との関連付けを行う。

(C05)過去のデータからの検討

診断履歴のデータベースが充実している場合には、経験的知識を統計的手法を用いて検証する。

(1-5) 知識表現の選択と適切さの確認

(C04)専門家の認識モデルに基づく知識表現

実装用の知識表現は専門家にとって理解しつらい場合が多い。知識獲得や保守の容易性のためにできるだけ専門家の表現に近い知識表現か望ましい。

(1-6) 知識獲得方法の選択と適切さの確認

(C01)専門家の選定方法の妥当性評価

エンジニアリングシステムの場合には保全者と設計者、医療診断の場合には隣接領域の専門医と言ったように、複数の立場の専門家からの知識獲得が必要とされる場合が多い。

(C06)知識の分類

診断の知識をモデル化し、知識の種類を適切に分類することにより知識獲得を効率よくしかも正確に行うことが可能になる。

(C12)テストケースの収集

診断知識だけでなくシステムの評価に役立てるためのテストケースを、知識獲得時に同時に収集しておくこと。

(2) 問題解決・推論方式の妥当性評価

実用的な診断型システムでは知識ベースが大規模になり、それにつれて問題

解決や推論方式も複雑になるため、妥当性評価が必要となる。経験的な診断知識には不確実さが伴うため、推論機構が不確実性を適切に扱っているかとうかを評価する必要がある。また、設計レヘルの知識を用いる場合には、定性推論や仮説推論などの手法の選択についても評価する必要がある。

(2-1) 推論処理の妥当性評価

(C01)試行錯誤的処理の必要性

デンジョンノリーを単純にたとるような場合はハイパーカートなとて実装する方が適切なこともある。エキスパートンステムのアプローチをとる必要性を確かめること。

(C04)質問のタイミング

実用的な診断型ンステムの推論処理は、単にデンジョンノリーをたどるだけではなく、診断の専門家の問題解決モデルを反映している必要がある。例えば点検コストを考慮して適切な質問の順序を定めるために質問制御の戦略に関する知識を利用することなとが有効てある。

(2-2) 推論時間の妥当性評価

(M02)推論エンジンの処理時間

推論時間が問題となった場合には推論エンジンの実装レヘルまでさかのぼって原因を追求し、必要に応して実装言語を変更するなどの対策が必要てある。

(M03)推論時間の増加オーダー

診断型ンステムは対象ンステムの例外的な状態を扱うのて診断精度を上げるためには知識ベースが大規模になる。利用形態として診断支援のために対話的に使われる場合が多いため、規模の増大に対する応答時間を評価する必要がある。

(2-3) 推論手法の選択と適切さの確認

(C01)問題解決のモデルやパラダイムの適切性

深い知識を用いる診断の手法として、Davisの制約留保、Reiterの論理に基つく診断理論、de Kleerの定性的推論などが提案されている。応用に応して適切な問題解決パラダイムを選択する必要がある。

(*C1)推論以外の方法を検討していること。

浅い知識を利用する場合には、従来型の手続き的な処理あるいはハイパーカートなどで実装した場合との比較評価を行う必要がある。

(2-4) 不確実情報に対する処理技法の適切さの確認

(C01)不確実性導入に伴う問題点

診断型問題ては経験的知識や観測データに不確実性が含まれる場合が多い。知識ベースの保守性を良くするためには局所的に定められる不確実性の指標と推論結果との関連を明確にする必要がある。特に知識ヘースが大規模化する場合には重要てある。

(C02)基本的な技術

診断知識に含まれている不確実性を取り扱うための手法として、MYCINて導入された確信度の方法、Dempster-Shaferの方法、ファジイ理論に基つく方

法、証拠推論などを比較検討する。

(C03)不確実性の局所化

利用者の立場からは不確実性を扱う必要がある場合でも、システムの内部ではできるだけ明確な意味付けのできる処理を行うことが望ましい。不確実さの度合を記号化するなどの方法を検討する。

(C04)不確実性の意味

観測データの記号化や記号の属性の記述など不確実性が発生するところをおさえてその意味を明確にすること。

(2-5) 学習機能の適切さの確認

(C02)学習機能の利用と問題点

学習機能の利用に際してはもとになる知識源（対象モデル、経験的知識、データなど）の持つ特性を考慮して問題が生じないかどうかを評価すること。

(M01)学習前／後の性能比較

(3) 結果の妥当性評価

設計型システムや計画型システムなどでは実行時に推論結果をある程度システムの中で評価できる。しかし診断型システムの場合には診断結果は修理などのアクションを通してしか評価できないのが通常である。このためシステムの評価段階で結果の妥当性について充分おさえておく必要がある。また、知識ベースの洗練へフィードバックする意味からも結果の妥当性評価は重要である。

(3-1) 正解の評価

(C01)正解に関するコンセンサス

正解でなかった場合の影響の大きさなどから例えば「故障原因の順位付けで第３位までの中に真の原因が含まれていること」といったように具体的な基準を定めておくこと。

(C04)オンラインデータ

プラントの診断などではテストのためにオンラインデータを利用できる機構を準備しておくこと。

(*C1)解に至った推論の道筋も妥当であること。

対話的に利用者を支援する診断型システムでは、解自身の妥当性だけではなく、どの様な過程をへて解に至ったかがシステムの信頼性の判断に大きな影響を与える。説明機能などを用いて評価することが必要である。

(M01)正解率の測定

予防診断的なシステムでは実際にプラントにおいて不具合を生しさせてテストすることができないためシミュレーションなどの代替え手段をとること。

(*M1)第１種の過誤、第２種の過誤の比率を測定すること。

真の原因を見過ごしたケース、および、真の原因でないものを原因と取り違えたケースの比率を測定する。

(3-2) 解の網羅性評価

(3-3) 解の最適性評価

(3-4) 解の評価方法の適切さの確認

(C01)評価方法

診断支援システムで最終的に操作の決定を専門の操作員が行うような場合にはフィールドテストが有効である。

(C02)解の性格

診断結果を誰がどんな場面でどのように用いるのか、そのためには解にはどのようなことが要請されるのか、等を明確にしておく。

(4) 要求仕様に対する達成度評価

ひとくちに診断型システムといっても実際に求められている機能は、不具合の原因を追求して設計へフィードバックすることや、専門家あるいはオペレータなどを支援して正常な状態へ早く復旧できるようにすること、など様々である。これらの要求に沿ってユーザのニーズを要求仕様としてまとめ、適宜達成度を評価する必要がある。

(4-1) 要求仕様内容に対する評価

(4-2) 要求仕様に対する達成度評価

(C01)第Ⅰ版要求仕様書

(C02)第Ⅱ版要求仕様書

(C03)実用版要求仕様書

診断型システムの場合は、システム化の目的と範囲とを要求仕様として明確にし達成度を評価する必要がある。また、実用版では知識ベースの増大が予想されるために品質・性能要件の達成度評価も重要である。

(C04)計画と実際の開発とのギャップ

(4-3) 要求仕様設定手段の評価

(5) 知識ベースの保守性評価

エンジニアリングシステムの診断では、対象システムの仕様が多様化しており、また仕様の変更も頻繁に起きるため、知識ベースの保守性が課題となっている。開発の初期から知識ベースの規模と運用方法とを想定し、保守性の評価を行う必要がある。

(5-1) 変更箇所発見の容易性の評価

(C02)推論処理に対応したモジュール化

診断の推論は、対象システムのモジュールや、不具合モードの階層構造などに対応づけてモジュール化される。このモジュール分割が適切に行われていることが知識ベースの変更箇所発見の容易性にとって重要である。

(C05)ツール、機能の整備

知識ベースが大規模化すると知識エディタなどの知識ベース保守支援機能なしでは変更箇所を把握することが困難になる。また、複数の部署から知識獲得を行う場合には用語辞書の充実が必要である。

(5-2) 知識ベースの理解性評価

(C01)専門家による保守

診断の専門家が理解しやすいように表や木構造などふたん利用されているやり方で知識を表現する。

(C04)記述レベル

実装レベルに近いルールではなく、より分かりやすい上位のルール表現を用意すること。

(C06)技術移転

知識提供者である専門家にも知識ベースの記述など関連するところは十分に理解がえられるようにすること。

(5-3) 変更箇所の局所性評価

(5-4) 知識ベースの統一性評価

(5-5) 品質安定性の確認方法の確認

(C03)バージョン管理

学習機能により自動的に知識ベースを変更する機能を持つシステムては、特にバージョン管理に留意する必要がある。

(5-6) 保守マニュアルの整備状況の確認

(5-7) 保守用ツールの整備状況の確認

(6) ヒューマンインタフェースの妥当性評価

支援的性格の診断型システムでは、利用者インタフェースが実用上特に重要視されるためその妥当性を評価する必要がある。

(6-1) 利用時のインタフェース評価

(C01)結果の表示

診断結果が緊急の対策を要するような場合には分かりやすい方法て操作員へ通知し、判断を助けるための補助的な情報も表示すること。

(C17)対話性

対話的な診断をスムースに行うためには、システムからの質問に逐一答える方式では不十分である。診断対象システムの点検結果やこれまての故障の履歴により分かっている情報を予めまとめて入力できることや、途中て入力の誤りを訂正して再推論できるなどの機能が必要である。

(6-2) 保守時のインタフェース評価

(C07)内部表現への変換

保守担当者が専門家であれば、フォーマットシートや表などの理解しやすい形式で知識ベースへの入力を可能にする必要がある。そのための内部表現への変換機能を持つこと。

(7) システムの総合評価

診断型システムは、測定システムや診断データベースなどとの統合をはかることにより実用性の高いものとなる。開発においては診断型に特化したツールがいくつか市販されており、適切なものを利用できれば開発効率の向上がはかれる。また、導入の効果を客観的に示す指標が定めにくいため開発の初期に充分詰めておかなければならない。これらの観点を含めた総合評価が必要である。

(7-1) テーマ選定の妥当性評価

(C01)ニーズの強さ

診断型システムは実験型プロトタイプのレベルまでは比較的容易に開発できるが、実運用をめざすシステムでは他のタイプと同様に十分なニースの検討が必要である。

(7-2) 利用者の満足度評価

(C01)現場に近い環境

診断型システムではマニュアルや熟練者など代替手段がある場合が多く、システム導入のメリット・ディメリットを現場で評価する必要がある。ただし、使い勝手など身近なところで評価される傾向があるため、システムの成長性や他のシステムとの統合化などを含めた評価とするよう留意する。

(C09)必要なデータ量

診断型システムではいかに少ない情報から正しい判断を下すかが利用者の満足度に大きく影響する。

(C10)出力結果の視覚化

点検・修理作業の支援に視覚化は大きな効果がある。

(C13)学習機能

システムからの応答が改善されていくとエキスパートシステムとしての利点が強くアピールされる。

(7-3) 統合性の評価

(C02)既存機能または既存システムとの連続性

計測システムや診断データベース、診断ネットワークなどのシステムとの連続性を評価する。

(7-4) 成長可能性の評価

(C04)プロトタイプ

プロトタイプを提示することにより、現実的な開発の方向付けが得られる場合がある。

(C06)開発環境と実行環境

効率的な知識ベースの開発と、現実的な推論時間とを両立させるためにはそれぞれの環境を準備することも考慮する。

(M01)資源の見積り

診断型システムでは知識ベースの充実によって取り扱う診断の範囲が広がり精度が向上する。しかし、知識の構造化が適切でないと必要資源の急激な増加を招く恐れがある。ライフサイクルの中で適宜チェックポイントを設け、必要に応じて基本構造を改良するなどの対策を行う。

(7-5) 開発／保守用トキュメントの評価

(7-6) 開発方法の適切さの確認

(7-7) ツール・言語選択の適切さの確認

(*C1)診断シェルの検討

診断型システム向けの領域特化ES開発ソール（診断シェル）がいくつか市販されている。対象問題がうまく当てはまればこれらを利用することにより開発工数の低減がはかれる。

(7-8) システム開発の生産性評価

(7-9) システムのコストパフォーマンス評価

(7-10)セキュリティ対策の評価

(C05)情報の開示レベル

診断情報は設計情報とも関連する機密性の高い情報であるため、システムの利用者に応じて診断情報をどのレベルまで開示するかを検討する必要がある。

3.5 5 制御型システム

3 5 5.1 制御型システムの概要

ここで言う制御型システムとは、制御の対象となるシステムに対して外部から与えられた運用計画および方針を基に、

- 時間・環境・システム状態の変化により直接操作または操作値を出力し（定常時制御）、
- 計画されたシステム状態からのずれが生じた場合にはそれを回復する操作または操作値を出力する（異常時制御）

エキスパートシステムである。

このタイプのエキスパートシステムの適用分野としては、

- 船舶・航空機・人工衛星等の運行・操縦制御
- 溶鉱炉・セメントキルン等のプラント制御
- 製造工程制御
- コンピュータハードウェア／ソフトウェアの実行制御

その他多くのものが考えられる。

現在でも何らかの自動／半自動制御がエキスパートシステム以外の手段で行なわれているものが多い。例として挙げたエキスパートシステムの適用分野を見ても明白なように制御システムおよびその中枢としてのエキスパートシステムが正しく動作することが絶対の条件になっている。エキスパートシステムの判断の誤り、誤動作は場合によっては人命にかかわったり社会的に大きな損失をもたらしたりする。従って、エキスパートシステムの検証および妥当性検査（V＆V）は必須条件である。制御型システムは入出力データが多く、しかも実時間性の複雑なシステムがほとんどであり、実際にはV＆Vの実施には多くの難しい問題を含んでいる。

3.5.5.2 制御型システムの特徴

制御型システムの特徴は、エキスパートシステムの基本タスクとしての分類ではなく、まず適用業務としての一分野としてとらえられるべきである。ここでは他の適用業務分野と比較して、特に制御型システムの問題領域に顕著な特徴をあげる。

(1) 対象システムの複雑さ

制御の対象となるシステムは個々にはかなりの違いはあるものの、問題のトメインは明確で比較的小さいものが多い。目標値あるいは目的関数の種類は限られており、その評価値も明確である。しかしながら対象システムは数多くのサブシステム・部分により構成され、お互いに干渉しながら全体として動作しており、それを制御するエキスパートシステムの内部モデルは複雑になる。対象システムの状況を表わす入力データの数は多く、場合によっては数千から数万のデータより状況を判断しなければならない。更に状況の判断には現在のデータ値だけでなく傾向値や周期的変動等の履歴データや事象・状況の発生順序による文脈判断が必要な場合も多い。制御値・操作の決定においてもサブシステム・部分の状況を判断し、それから全体としての状況判断に至る総合化機能の実現、判断より個々

の操作値・操作および操作順序を決定する分解・組立て機能の実現が必要である。すなわち制御型システムはそのサブシステムとして解釈型／診断型／設計型／計画型のシステムを含む複合型システムである。

(2) あいまいさおよび非単調性

対象システムは時間、環境の変化、事象の発生等により急速に変化する。エキスパートシステム内部に保持しているデータやモデルは、それにつれて急速にデータとしての正確さを失ってゆく。新しく入力されたデータは、モデル内部のデータと相いれない場合には、その内部データに依存した生成知識は更新されなければならない。エキスパートシステム内における制御知識は、ほとんどがある特定の状況でのみ成立するものである。また入力データそのものも必ずしも正確ではなく、時間遅れ、計測誤差等によるあいまいさを含む場合が多い。そのような場合でも安定した制御を限られた資源、時間内で行う必要がある。

(3) 実時間性

対象システムの性格により、かなり差異はあるが、制御型システムは通常実時間性が厳しく要求される。システム状況はダイナミックに急激に変化し、素早い操作を行わないと対象システムが動作不能になったり制御不能になったりする場合もありうる。継続して起こる多量の変化する入力データに対して正確な制御値・操作を許容される時間内で決定しなければならない。応答時間は、数ミリ秒の厳しいケースから数分の比較的に軽いケースとかなり差異はある。また応答を返さなくても対象システムが通常安定稼働するものと、限定された時間内に応答がないと対象システムが停止する場合ではシステム要件はかなり異なる。しかし、他の問題領域システムよりは非常に強い実時間性が要求される。

(4) 高信頼性と精度の向上

先に述べたように制御型システムの誤動作は大きな損失をもたらす。出力としての制御値・操作の決定には非常に高い精度が要求される。さらに制御型システムでの精度には実時間システムとしての要件がある。精度の高い出力を行っても、それが要求された時間内の応答でない場合には役に立たないばかりでなく、時には障害を引き起こす可能性もある。すなわち制御型システムにおける出力の精度は時間と質で決定される。質があまり高くなくても一定時間に応答できるシステムは、質がより高いが一定時間に応答できないシステムより実用的である場合が多い。

制御型システムの場合にはエキスパートシステムの精度を向上する仕組を比較的作成し易い場合が多い。トランザクション処理としてのエキスパートシステムの性格により多数の推論ケースを得ることができる。エキスパートシステムの妥当性検査においては、推論経過の記録と実制御システム（専門家そのものの場合も有りうる）との比較、または確立した評価方法によりテストは比較的実施しやすい。ファジィ制御、ニューラルネット、事例ベース推論等を使用している場合には過去の推論の統計値の使用等により自動的に精度を向上することも可能である。

3553 制御型システムの基本タスク

前項の特徴および問題点に対処するための制御型システムエキスパートシステムの基本タスクを次に説明する。

(1) 複雑さに対するエキスパートシステムの基本タスク

①入力データ解釈および内部モデル更新
フレーム、セマンティックネット、オブジェクト指向等の知識表現方法を用いて対象システムの状態をエキスパートシステム内部に表現する。

②履歴・文脈判断
時系列推論、状態遷移ネットワーク等の手法を用いて対象システムの状況を把握する。

③複数知識ベースの構成と実行
サブシステム・部分の状況を判断する複数知識ベースと総合判断知識ベースの同時実行を可能にする仕組と相互コミュニケーション機能を提供する。黒板システム等がその例である。

④個々の制御値・操作の生成
制約、システム動作、過去のケース等を用いて制御値・操作を決定する。

(2) あいまいさおよび非単調性に対するエキスパートシステムの基本タスク

①あいまいさを取り扱う知識表現
ファジィ、ベイス等の手法を用いて対象システムの広い動作範囲において安定した制御がコンパクトな知識ベースで実現できる。

②非単調推論
エキスパートシステム内知識のダイナミックな更新、ATMS等の手法を用いて常に対象システムの正確なモデルを保持できる。

(3) 実時間性に対するエキスパートシステムの基本タスク

①タイマ機能
応答時間を保証するためには、応答を要求するような事象が発生してからの経過時間を計測するタイマが必要である。必要とする応答時間は事象によって必ずしも一定てはない。また幾つかの推論が同時平行して実行される場合には複数のタイマが設定できなければならない。時間の進行により推論のサイクルに割り込める仕組が必須である。

②推論の変更
一定時間内に必ず応答が要求される場合には推論の途中での応答可能性をチェックする場合がある。そのままては時間内に応答を生成するのが不可能

な場合には途中で推論方法を変更し、以前の事例を用いたり、簡便な知識ベースで最低限の精度の応答を生成する。

③優先順位処理

エキスパートシステムが推論中に対象システムに重要な事象が発生し、その応答を生成するのが急務となるケースはよく発生する。優先順位に基つく推論の一旦停止／優先推論の実行／もとの推論の再開等の仕組が必要てある。

④高速推論・並列推論

必要とする応答時間内で精度の高い制御値・操作を決定するには限られた資源内で高速推論を行う仕組が必要である。場合によっては複数知識ベースを用いて平行推論を行う場合もある。

(4) 高信頼性と精度の向上に対するエキスパートシステムの基本タスク

①安定した制御値・操作の保証

エキスパートシステムが出力した結果の精度をチェックし、対象システムの操作許容範囲内てあることを保証する。これには時間的制約も含まれる。場合によってはエキスパートシステム手法ではなく動作範囲が確定している従来手法による二重チェックも用いられる。

②推論途中結果の精度計測

何らかの精度評価関数の設定ができる。あるいは推論結果を改良してゆく推論方式を採用する。

③推論の過程・結果を記録・利用

結果の精度を評価する方法を確立し、それによりエキスパートシステム内の知識ベースを自動的に更新することができる。

④高使用可能性

制御型システムは長時間にわたる継続した安定稼働が原則てある。エキスパートシステム自身の安定稼働は勿論のこと、障害時／誤動作時のバックアップへの切り換え／回復機能が重要である。その場合にはエキスパートシステム内の知識ベースおよび対象システムのモデルのバックアップエキスパートシステムへの転送の仕組が確立していなければならない。あるいはエキスパートシステムを二重化する場合もある。

3 5 5 4　制御型システムのV & V

ここにおける制御型システムとは、主に全自動型の制御型エキスパートシステムを言い、全体の制御システムに組みこまれたエキスパートシステムを取り扱っている。操作員がエキスパートシステムに制御データを求め、それを制御システムに手入力するオフラインの制御型エキスパートシステムはここては取り扱わない。また制御フィードバックの時間に対する要求も数秒から場合によっては1秒以下の実時間性の強い制御型システムを想定している。以下の点検項目・測定項

目は、すべて以上の状況を前提として説明されている。

(1) 知識ベースの妥当性評価

制御型システムでは、知識ベースの妥当性が全体システムの出力品質に直接影響する。知識ベースには制御の方針・論理・手順が収容されるので、単体の知識の妥当性よりも組み合わされたサブシステム全体システムとしての知識ベースの妥当性がチェックされなければならない。

(1-1) 知識ベースの正当性評価

(*C1)モデル選択の正当性

制御型システムにおけるモデルおよび制御知識の選択においては、必ずしも専門家が現在使用しているモデルにこだわる必要はない。エキスパートシステム化による処理の高速化により、現在のモデルより大きなあるいは多くのモデルを使用することが可能になる場合も多い。要は結果としての出力品質にどれだけ寄与するか、実用的であるかが決め手になる。

(*C2)モデルの適用範囲

どのような制御モデルであっても、その適用には入力値限界・適用制約条件があり、特定の空間・時間・状況での正当性しか持っていない。

従って制御システムでは通常複数のモデルを組合せて使用する。想定される制御システムの適用範囲全体において、どの条件において、とのモデルを使用するかを詳細に検討する必要がある。

(1-2) 知識ベースの完備性評価

(C01)知識の分解レベル

知識として蓄えられている個々のモデルの適用制約条件とエキスパートシステム全体としての制御範囲は必ずしも一致しない。制御システム全体の入出力限界およびその他の制約条件について、対応するモデルが欠落している状況が発生しないか、複数のモデルが適用可能な場合にはどのモデルを選択すべきかをチェックする必要がある。

(1-3) 知識ベースの一貫性評価

(C04)知識の断片性

状態遷移ネットワークなどの手法を用い、制御対象システムの発生事象・時間による変遷をまとめ、制御知識の重複・欠如をチェックする。

制御システムは、通常入力データが多種多量になる場合が多いので全ての条件組合せをチェックすることは不可能である。代表的な事象の発生に沿って使用される知識・モデルおよび出力をチェックして、急激な制御出力値の変化がないかをチェックする。更にそれが妥当なものかチェックする。

(C07)ルールの一貫性のテスト

制御知識はその適用条件によりサブグループ化される。サブグループ内では知識間の矛盾はないはずである。

(1-4) 経験的知識を正当化する方法とその確認

(C03)統計値を利用した正当化

制御システムのエキスパートシステム化において、全体の枠組のデザイン

決定については、専門家の知識が主たる正当化の手段であるか、個々の制御知識については、適用限界あるいは入力条件と制御出力との相関などにおいて、豊富な統計データが利用できる場合が多いので、積極的に利用すべきである。

(1-5) 知識表現の選択と適切さの確認

(C04)専門家の認識モデルに基づく知識表現

制御知識を知識ベース内に実現する場合に専門家の認識しているモデルとあまりにもかけ離れた表現方法を取ると知識獲得・保守・更新が困難になる。また妥当性の評価ができない。

(C05)知識表現の効率性

入力データが多種多量で時間経過とともに急激に変化する場合、それをモデル内で効率よく表現する仕組が必要である。制御対象システムの変化に追随できずに制御出力が発散する場合がある。制御モデルは表現力とともに簡潔で効率の良いものを選択することが重要である。

(1-6) 知識獲得方法の選択と適切さの確認

(C03)知識獲得の網羅性

制御型エキスパートシステムは、他のエキスパートシステムと異なりその実行時に人間系の支援を期待できない場合が殆どである。したがってその知識ベースは完結していなければならない。知識は必ずしも専門家から獲得されるものだけではなく、制御における常識といったものも必要である。

(C08)知識獲得支援方法

制御対象システムの行動をモニターして、各制御時点での必要知識の確認、その知識源を確認するウォークスルーが有効である。

(2) 問題解決・推論方式の妥当性評価

適切でない問題解決・推論方式を選択すると、制御の方針・論理・手順を表現しにくかったり、開発・保守の生産性が低下して、制御システムにエキスパートシステムを使用する利点はなくなってしまう。更に実時間での応答が不可能になり、システムとして作動しない危険もある。

(2-1) 推論処理の妥当性評価

(C02)メタ知識

制御対象システムの個々の状態に対応する制御知識群と、対象システムの変化に合わせて適切な知識群を喚起するメタ知識を明確に区別することが必要である。個々の制御知識群は、保守あるいは学習により頻繁に変更される場合が多いが、メタ知識は全体の制御戦略となるため変更されることは少ない。

(2-2) 推論時間の妥当性評価

(C03)タイマによる監視機能

制御エキスパートシステムを実時間システムとして使用する場合、推論時間についての要求は厳しいものになる。システムの実時間性とは、

- 事象の発生に対する即応性
- 一定の時間内あるいは一定の時間経過時に応答を返す即時性
- 優先順位に基つく多元事象のスケジューリング機能、これにはプリエンプティブなスケジューリンクも含む

などの要件を満たすことてある。制御エキスパートシステムにおいても短かい時間で結果を出力するたけでは実時間ンステムとは言えない。
すなわち、

- 事象の発生と同時にその要求応答時間を決定する。
- 応答を必要とする個々の事象に対して経過時間を監視する。
- 推論時間の監視のためにタイマ待行列を設定し推論に割り込む。
- 時間の経過に従い、場合によっては推論方法を変更して応答時間を守る。
- より優先順位の高い事象の発生に対して、場合によっては現在の推論を中断しその事象の処理を優先する。（プリエンプティブなスケジューリング）

制御ンステムにおいては、対象ンステムの状態が急激に変化したり、周期的に変化するために要求される応答時間（推論時間）は一定てはないことに注意する必要がある。

(2-3) 推論手法の選択と適切さの確認

(*C1)即応性

前述の実時間性により、推論結果の正確さよりも即応性が優先する場合が多い。すなわち、より正確な結果が得られるが、許容推論時間を超えるような手法よりも、許容推論時間内でそこそこの結果を得られる手法か実用的である。例えば、正確さのために全ケースの探索により最上の結果を出力するよりも、事例ベース推論により以前の実績を利用して応答性と精度を確保する場合も多い。いずれにしても時間を考慮した効率よい推論手法が求められる。

(2-4) 不確実情報に対する処理技法の適切さの確認

(C02)基本的な技術の検討

制御型エキスパートシステムにおいては、連続量の取扱いについての統一的な推論手段としてファジィ理論が有効な場合も多い。しかしなから、メンバーンップ関数の決定については、制御対象ンステムの観察・測定・統計処理によりその根拠を明確にしておかねばならない。特に多段の推論を行う場合には、急激にその信頼性を失う場合が多いのて注意が必要てある。

(2-5) 学習機能の適切さの確認

(C01)実現可能な基本技術の検討

制御型エキスパートンステムにおいては、入力テータ・推論処理経過・出力データおよびその結果としての制御品質データか多数事例として自動的に得られる。それらを用いて制御知識を自動更新する仕組を作成し易い。そのためには、先に述へたように制御知識とメタ知識の明確な分離が必須である。さらに多くの事例を効率良く蓄積し利用できる支援サブンステムの設計が必要てある。

(3) 結果の妥当性評価

制御システムの出力品質は、全体の業務プロセスへの損益に直接影響する。何らかの方法で出力品質を測定することがシステム改善の手掛りであり、大きな利益を生み出す機会にもなる。

(3-1) 正解の評価

(C04)テストデータ収集機構

制御型エキスパートンステムては通常多量の入力データを取り扱うため、試験的な実行時には実システムよりデータを取り入れ、専門家による制御との平行ランによる比較を行うのが普通である。

(3-2) 解の網羅性評価

(3-3) 解の最適性評価

(C02)評価関数の設定

制御型エキスパートンステムでは結果出力が制御値として制御対象ンステムにフィートバックされ、制御対象システムの性能・生産性を通じて最適性が判断される。制御対象ンステムの生産性の定義は通常行なわれているはずである。制御型エキスパートンステム採用以前と以後の性能・生産性比較により相対尺度としての最適性が定義される。個々の入力テータと結果出力のサイクルにおける最適性は定義しにくい場合か多い。

(3-4) 解の評価方法の適切さの確認

(C01)評価方法の検討

制御型システムでは入力データ、状況データ、結果出力データ、および生産性評価データ（あるいはその算出のためのデータ）が採集可能てある。制御型ンステムにはそれらのデータをログとして出力する機能か備わっていることが望ましい。得られたログより、生産性測定のための性能統計値を抽出する。生産性評価のための尺度は制御対象システムの特性により様々であるが応答時間および信頼性、特に強固性(Robustness)は重要な評価尺度である。

(4) 要求仕様に対する達成度評価

制御システムにエキスパートシステムを用いるのは、他の方法と比へて何らかの優位性、例えば開発・保守の生産性、結果の品質向上等の期待があるからである。要求仕様にはそれらが含まれるへきてあり、それに従って評価されなければならない。

(4-1) 要求仕様内容に対する評価

(C01)正当性

制御型エキスパートンステムは、他のタイプのエキスパートンステムの要求仕様の内容と異なるわけてはないが、目標とする性能・生産性の向上、および信頼性、応答時間については通常必す明記されており客観的な数値により達成度を評価すへきてある。

制御型エキスパートンステムにより自動制御される対象ンステムの範囲お

よび制御レベルについてはよく吟味し、仕様として明記しておかないとユーザ部門の持つ期待とのずれが問題を引き起こすので注意が必要である。

(4-2) 要求仕様に対する達成度評価

(*C1)達成度の尺度

制御システムにエキスパートシステムを採用する動機としては次のようなものが考えられる。

- 連続／非連続データの双方を一意に取扱いたい。
- 多数の制御モデルがそれぞれの制約条件内で動作し、全体としての統一的な制御システムにするのが従来手法では難しい。
- 制御知識が専門家のカン・経験でしか存在しない。
- 制御システムの保守作業を軽減したい。
- 過去の事例を制御に生かした効率のよい制御を行いたい。

いずれにおいても、エキスパートンステムとしての利点が充分に発揮できずに、ただ単にエキスパートシステムツールを使用したたけのンステム開発は従来の手法を試みるへきである。

要求仕様内容に対する評価で述べたように、達成度は数値により評価すへきである。

(4-3) 要求仕様設定手段の評価

(5) 知識ベースの保守性評価

制御型システムにおけるエキスパートシステムの利点は、特に知識ヘースの保守性の飛躍的な向上である。専門家が持つ制御知識を理論やアルゴリスムに変換することなく直接に表現できる利点は大きい。知識ヘースの保守性を防げるような設計は、せっかくの利点をなくすることに等しい。

(5-1) 変更箇所発見の容易性評価

(C01)モジュール分割とシステムの外部機能との対応

制御型エキスパートシステムのモジュール分割には様々な方法か考えられるが、保守性から考慮する場合、制御対象ンステムのサブンステムおよびその制御条件、状況・時間等により分割すると良い結果が得られる。

(C04)トレース機能

制御型エキスパートシステムでは多数の事象がランタムに発生し多数の制御知識が動作する。保守のためのトレースは必須である。オーバヘットの少ないトレースを常時実施するようにしておきたい。

(5-2) 知識ベースの理解性評価

(C05)見通しのよい知識構造

多数の制御知識を制御対象システムの状態遷移ネットワーク（例えばATN、Petrinetなど）などの手法により分類し、全体として見通しのよい構造を持っていること。実際の制御知識とメタ知識が明確に分類されていること。

(5-3) 変更箇所の局所性評価
(C01)モジュール間のインタフェース
(5-1)および(5-2)参照。

(5-4) 知識ベースの統一性評価
(C02)モジュール分割の観点の一貫性
(5-2)参照。

(5-5) 品質安定性の確認方法の評価
(C02)動作確認の仕組
制御型エキスパートシステムでは実行時に人間系の支援を得ず独立して動作する場合が多いので、その安定性については充分なテストが必要である。現在の専門家による運転時に入力データを採集し、制御型エキスパートシステムにフィートしてテストを行ったり、専門家との平行運転によるテストを行う。安定性を確認できる業務サイクルに従ったテスト期間・量が必要である。更に安定性を求められるシステムでは保守時においても本番エキスパートシステムと保守済エキスパートンステムとて平行運転を行ない、安定性を確認したあとで本番エキスパートシステムと切り換えるなとの措置が取られる。

(5-6) 保守マニュアルの整備状況の評価

(5-7) 保守ツールの整備状況の確認

(6) ヒューマンインタフェースの妥当性評価
制御型システムでは直接の利用者は人間系ではなく、その意味てのヒューマンインタフェースの考慮は不要である。システムの開発・保守あるいは妥当性評価のためのヒューマンインタフェースを考慮する必要がある。

(6-1) 利用時のインタフェース評価

(6-2) 保守時のインタフェース

(7) システムの総合評価
制御型システムは単独で使用されることはなく、制御プロセスの構成要素として働く。システム全体での総合評価が要素としてのエキスパートシステムの評価でもある。

(7-1) テーマ選定の妥当性評価
(4-2)参照。

(7-2) 利用者の満足度評価

(7-3) 統合性の評価

(C02)他システムとの連続性

制御型エキスパートシステムは信頼性において完全ではない。エキスパートシステムがコントロールを失った場合には、速やかにシステムを安全な方向に導いたり、人間系にコントロールを譲り渡したりできるようになっていなければならない。エキスパートシステムからの制御出力を直接対象システムにフィードバックせず、従来手法を用いたシステムにて最終チェックして対象システムに入力することも考えられなければならない。

(7-4) 成長可能性の評価

(7-5) 開発／保守用トキュメントの評価

(7-6) 開発方法の適切さの評価

(7-7) ツール・言語選択の適切さの確認

(C03)ツール・言語の選定

実時間システムとしての性格、多量のデータ処理としての性格からツールは高性能であること。制御システムの一部としてエキスパートンステムは稼働するので従来手法との組み込み容易であることが必須てある。

(7-8) システム開発の生産性評価

(C03)生産性目標は妥当であること。

前述のように制御システムにエキスパートシステムを使用する動機には開発・保守の生産性向上がある。従来手法による実績あるいは見積との比較を行っておくこと。

(7-9) システムのコストパフォーマンス評価

(7-10)セキュリティ対策の評価

3.5 6 複合型システム

3 5 6 1 トメインの特徴

(1) 複合型ンステムの定義

複合型システムは、解析型問題(analysis problem)と合成型問題(synthesis problem)が混合したシステムとして定義できる。従って、解析型問題と合成型問題の特性を合わせ持つシステムである。制御型システムと非常に似たところがあるが、ここでは、知的ナビゲーションシステムや知的ＣＡＩシステムなとに見られるように、ある時点の状態を分析し、その状態に対して最適な解（プラン）を見つけだすシステムとして複合型システムを定義する。

つまり、まずシステム自身の内部状態、あるいは外部からの情報を分析し、対象物の状態を推定する。その推定結果を基に、決められた拘束条件のもとで、所与の要件を満たす解を組み上げ、実行するタイプのシステムである。

制御型システムとの対比を明確にするために、知的ＣＡＩンステムを具体的な事例として、特徴、問題意識、Ｖ＆Ｖについて述へる。

(2) 複合型システムの実現手法

基本的には典型的な解釈型問題である解釈型システムと典型的な合成型問題である計画型システムの特性を受け継いでいるが、それぞれの特性はかなり異なる。異なる特性を合わせ持つことが複合型システムの特徴となる。

知的ＣＡＩシステムでは、生徒の理解度を把握する部分が解釈型問題であり、その結果に基づいて教授戦略を立案するのが合成型問題として捉えることがてきる。その詳細を以下に述へる。

生徒に色々な質問をすることによって、その生徒の理解モデル（一般的には誤答モデルが用いられる）と比較することによって、その生徒の理解度を分析するとともに、どこが理解出来ていないかを推定する。従って、一般的に知的ＣＡＩシステムでは、教師モデル、生徒モデルなどの対象（人間の思考）に関するモデルを利用する。しかし、真の誤りや生徒の理解度を的確に把握するのは非常に難しい。なぜなら人間の思考を外的な観測情報から推測するためて、得られる情報が表層の情報であり、真の誤りはその裏側に隠れている場合が多いからてある。

このように非常に多くの不必要な情報や、重要な情報が欠落した不完全な情報から真に必要とする情報を探す（推定）のは非常に難しい。さらに、不完全な情報に基づいて状態を推定すれは、不十分な状態推定となる。不完全な情報に基づく解釈型問題であり、対象モデルの記述と共に曖昧性を含んた問題の特徴を充分理解する必要がある。

次に、推定された状態に対して、最適な教授戦略を立案する必要がある。教授戦略としては、生徒の能力と教授目標に沿った適切な学習プランを立案し、その計画に基づいて最適な教育を実施する。生徒の能力推定は、不十分な状態推定から行われるのでかなりの曖昧性を含んでいる。したがって、策定されるへき学習プランの問題空間は、ベースとなる生徒の能力推定があやふやなので問題空間自体もかなりの曖昧性を持つことになる。教授目標は比較的明確なので、現在の理

解度と生徒の能力から教授目標を達成するための学習プランを立案するのはさほど難しくはない。多くの場合、学習カリキュラムとして何パターンかを用意し、その中から最も良さそうな学習カリキュラムを選択することになる。このことは、既存計画の再利用に相当する。選択された学習カリキュラムが適切か否かを評価するのは難しい。人間の思考モデルと物理的な学習カリキュラムとの総合評価となる。その評価結果によっては、学習カリキュラムを動的に変更する必要もある。

3 5 6.2 タスクの特徴

複合型システムのタスクの特徴として以下の項目が考えられる。

(1) 対象のモデル化

知的ＣＡＩシステムを例にとると、モデル化する対象には、学習者側の理解（誤答）モデル、教師側の教授モデルなどがある。学習者の理解度を正確に把握して、何が理解できなくて、どこをどのように間違ったかを推定することが知的ＣＡＩシステムでは非常に重要となる。

従って、学習者の理解度をベースとしたモデル化が行われ、そのモデルとシステムが持っているモデルとの比較検討によって学習者の理解度を把握する。対象のモデル化には色々な手法が採られるが、解釈型問題で最も一般的なのが分類階層による方法である。分類階層でモデル化できれば、問題解決機構としては、分類階層に沿った探索手法を用意すればよい。

しかし、大規模探索問題では、単純な分類階層では組み合わせ的爆発を防くことができない。また、複雑な問題では、単純な分類階層ではモデル化が困難な場合が多い。本質的には、深い知識による対象のモデル化が必要となる。

(2) 不完全な情報に基つく対象の状態推定

学習者の学習状態を把握する場合、推定するための全情報を収集てきない場合が多い。そのような場合、システムは会話しながら必要な情報を収集しようとするが、“わかりません”という回答に頻繁に出くわしてしまう。

このような欠落した情報や不明瞭な情報から状況を把握する場合があるので、曖昧さを処理する機能が必要となる。最も似たモデルを探すためには、少々の不完全性を克服できるモデルの曖昧検索機能が必要となる。曖昧検索を実現する方法として類似検索がある。

類似検索は、与えられた対象（モデル）の間に成立する類似度を見出し、その類似度を利用して、不完全な情報から似たモデルを捜し出す方法である。

(3) 解釈の多様性

曖昧さを許して状態を推定すると、その推定結果を解釈する上で多様性が生しる。その多様性を処理するには、学習者に曖昧さを解消するための適切な質問を行う方法が一般的にとられる。

学習者の理解度が推定されたならば、理解度を増進させるための教授戦略を立案する。そのための基本タスクとして、まず、学習者の理解状態の推定結果から、教授戦略立案のための制約条件の作成を行う。作成された制約条件は、相互の関連性を無視して作成される場合が多いので、作成された制約条件間で矛盾があったり、教授戦略の立案が不可能な制約条件が作成される場合がある。それらの制

約条件の相互作用を処理するタスクが必要となる。

(4) 組み合わせの爆発を防く推論

最終的に制約条件が決定したら、合成型問題と同様な組み合わせの爆発を防く手法での探索タスク（推論機能）が処理をする。組み合わせ的爆発を回避する方法として、探索空間を階層的に分割した階層的探索手法がとられる。

階層的探索手法で一般的に利用される手法は、階層的生成検査法てある。階層的生成検査法は、『システムの構造を階層的に表現し、部分構造を段階的に生成し、その妥当性の評価を行って、有効てはないと判断される部分構造を排除しながら、最適なシステム構造を見つけだす』［Kobayashi 1986］とする手法てある。

組み合わせ的爆発を回避する別の方法に、トップダウン詳細化戦略がある。この方法は、『まず詳細な部分を無視して、抽象度の高い水準での粗い計画を立て、次により詳細な部分を考慮に入れて計画を詳細化する過程を繰り返しながら、最適なシステム構造を決定していく』［Kobayashi 1986］手法である。

この方法では、高レベルの探索空間を小さいものに限定し、かつその解が次のレベルでの探索に対する制約条件となる。小さいものに限定された探索空間を副問題として捉え、副問題の目標をサブゴールとして定義すると、サブゴール間の相互作用が制約となり、制約をシステムが達成する目標の部分記述とみなすことができる。

各副問題ごとに局所的に何らかの決定の行える情報が利用可能な場合、その副問題ごとに完全な解を与えないで、最小限の部分的な解の記述に留める。このように、決定の一部を先送りにする方法を制約条件の最小化という。問題の分割化をおこなうために有効な手段となる。

問題を副問題に分割し、制約を副問題のサブゴール間の相互作用として推論していくには、推論の過程てのサブゴールを評価しながら推論していく必要がある。一つの副問題で複数のサブゴールが生成される場合には、最良優先探索なとの手法で解を生成していく。

ある制約条件の下で、解が生成されたとして、その解が全体として最適な解とは言えない場合がある。解全体の整合性をとりながら、より良い最適解を捜し出す方法に焦点調整法がある。

焦点調整法は、連結されたサブゴールの集合として解を捉えれば、全体の解の中で不具合／不適切なサブゴールを捜し出し、その不具合／不適切な状態を解消して、より良い解とするために複数のサブゴール間でサブゴールの交換を行うものである。この交換は、当然、より良い解を生成する場合に交換が成立する。焦点調整法を適用する場合に、作成した計画案の評価法が重要てある。

焦点調整法で用いられる一般的な評価法は、得点法てある。各々のサブゴールを評価するために、問題の特性をよく捉えられる方法て評価関数を決める。その評価関数に従って、各サブゴールの得点を計算する。焦点調整法ては、その得点の低いサブゴールから交換していくことになり、交換によってサブゴールの得点が高くなる場合に交換が成立する。この交換は２つ以上のサブゴールても成立させる必要がある。

3.5.6.3 複合型システムのV&V

(1) 知識ベースの妥当性評価

複合型システムでは、対象の状況を捉えるための解釈部の知識ベースとその結果を制約として作成される計画部の知識ベースを分けて評価する必要がある。

知的CAIシステムの例では、解釈部の知識ベースは、生徒の理解モデルが中心となる。理解モデルは、人間の思考のモデル化、対象領域の問題（教材）に関する知識のモデル化が重要となる。特に、人間の思考のモデル化は、人間の思考自体、構造が非常に複雑であり、且つ、かなりの曖昧性を含んでいるため、非常に難しい対象である。

一般的には、理解モデルの代わりに誤答モデルが使われる。誤答モデルにおいても対象が難しいので、知識ベースの評価が重要となる。曖昧性を含んだ知識、特に浅い知識を前提としたシステムでは、知識の欠落が頻繁に起きる可能性があるので、知識ベースの完備性が重要となる。計画部の知識ベースでは、解釈部で決定された制約が曖昧性を多分に含んでいる可能性があるので、間違った計画案を作成する可能性がある。それらを防ぐための知識など、不完全性を前提とした知識ベースであり、妥当性を評価するのは非常に難しい。

(1-1) 知識ベースの正当性

(C01)整理された知識は、知識源を遡りその出所を確認すること。
　　特に，テスト等で得られた誤答の知識は、充分に吟味する必要がある。

(C05 ～C07)知識の解釈が異なる時，専門家間での合意をとること。
　　誤答を解釈するときの多義性，曖昧性の問題を検討する。

(*M1)正当性を検証するための定量的評価を実施すること。
- 適切な誤答モデル／誤りデータ
- 適切な教授戦略／誤答モデル

(1-2) 知識ベースの完備性

(C01 ～C02)知識の分解レベルの詳細度がアンバランスでないこと。
- 個々の誤答モデル、教授戦略のレベルの評価。
- 知識構造のレベルの検討。
- 誤答モデル等に使う知識の粒度の整合性の検討。

(*M1)誤りデータ数の評価とモデル化の正当性を定量的評価に評価すること。
- 誤りの種類数／充分な数の誤りデータ数
- 誤答モデルの数／誤りの種類数
- 教授戦略の数／誤答モデルの数

(1-3) 知識ベースの一貫性

(C08 ～C10)知識（ルール）の適用範囲について検討すること。
- 重複した誤りに対する検討。
- 単純ミスと真の誤りとの分離の検討。

(C12)知識の変更に対しグローバルな整合性が保てること。
　　誤答モデルのモデル追加により，知識間の矛盾が生じていないか。

(*M1)一貫性を検証するための定量的評価を実施する。
- 競合知識／誤りデータ

・競合解消のための質問数／競合知識

(1-4) 経験的知識を正当化する方法とその確認
(C02)深い知識によるモデル化を検討すること。
誤答モデルを深い知識で表現が可能であるかを検討する。

(1-5) 知識表現の選択と適切さの確認
(C01)各知識表現の得失を整理すること。
複合型では，複数の知識表現を融合させて使用する場合が多いので，知識表現の融合を考慮した知識表現法の選択が必要である。
(C07)保守が容易な知識表現法であること。
誤答モデル、教授戦略の追加は容易か。

(1-6) 知識獲得方法の選択と適切さの確認
(C09)知識の精粗のレベルを統一すること。
誤答モデルのレベルは、不統一になりやすいので特に注意する。

(2) 問題解決・推論方式の妥当性評価
複合型システムでは、解釈部の推論の結果を基に計画部が教授戦略を立案する。その結果に基づいた教材を学習者に呈示し、その反応を基に解釈部が学習者の状態を推論する形となる。このように人間とシステム、システム側でも解釈部と計画部が連なった形の推論が行われるので、全体の整合性のとれる問題解決・推論方式である必要がある。

(2-1) 推論処理の妥当性評価
(C02)専門知識、推論戦略知識、推論エンジンの分離が明確なこと。
誤答の知識は追加、削除、更新が頻繁に発生する。それにより、全ての知識の構造等が変更される可能性がある。
(*C1)推論で利用される知識及び中間結果が妥当であること。
生徒に理解させるために説明機能が必須となる。その為に推論過程て生成される仮説は適切でなければならない。
(*C2)知識の順序変更に対しても同じ回答が出されること。
頻繁に知識が変更されても、知識ベースの頑健性を保証する必要がある。
(*M1)推論の妥当性の定量的な評価を行うこと。
・最長と最短の推論段数の比

(2-2) 推論時間の妥当性評価
(*M1)適切な推論時間であること。
・応答時間（会話型のため、２～３秒以内）

(2-3) 推論手法の選択と適切さの確認

(2-4) 不確実情報に対する処理技法の適切さの確認
(C03)不確実性の局所化を検討すること。
不確実な場合には，質問等により明確化する手法を考慮する。

(2-5) 学習機能の適切さの確認

(3) 結果の妥当性

知的ＣＡＩシステムでは、結果の妥当性の評価が極めて難しい。その理由は、誤りの原因が学習者自身にあるため、開発者側だけでは結果（誤り原因の同定）の正当性の判定が行えないからである。特に、学習者の誤り方も千差万別であるためにＫＥと専門家では、完全なる結果の妥当性評価が行えない。そのため、実際に結果の妥当性を評価するには、業務に適用しながら妥当性を評価していく必要がある。

(3-1) 正解の評価

(C01)正解の定義をしておくこと。
　　学習者が理解の誤り原因を正確に認識できるカイトを出せるか。

(C06)解は，本当に正解か否かの吟味をすること。
　　似ているが間違ったカイトが出力されていないか。

(M01)正解率を定量的に把握しておくこと。

- 正解率＝適切なガイト出力数／総テスト問題数
- 適切なガイト率＝適切なガイト数* ／総テスト問題数

　*:「正しい」との指摘も含めた適切なガイド出力数

(3-2) 解の網羅性評価

(C01)解の単一性，複合性など，網羅性の目標を充分検討すること。
　　複合型システムでは，知識ベース間にまたがって推論されるため，特に注意する必要がある。

(*M1)用意した全テストデータに対し実施する。

- 適切な教授戦略／テストデータ数

(3-3) 解の最適性評価

(3-4) 解の評価方法の適切さの確認

(*C1)学習者のレベルに応した解かどうかの検討をすること。

- 学習者の理解度に合ったガイトが出力されているか。
- 理解度が低い学習者でも理解できる内容が出されるか。

(4) 要求仕様に対する達成度評価

知的ＣＡＩシステムにおける要求仕様は、対象とする学習者の理解能力（出来る子、全然出来ない子など）に応した最適教材の呈示による最大学習効果にある。

このように目的意識の比較的明確なシステムでは、要求仕様に対する達成度の評価は絞り込みやすいが、最大学習効果など抽象的な評価項目を定量的に捉えるのは非常に難しい。

(4-1) 要求仕様内容に対する評価

(*C1)要求仕様の記述に誤りや脱落がないこと。
　　目的（思考支援／学習支援等）、実現の機能が明確に書かれているか。

(4-2) 要求仕様に対する達成度評価
(*M1)ユーザ要件が達成されているかの定量的な評価を行うこと。
• 性能要件（応答時間、保守性等）
(*M2)要求仕様の達成率の定量的な評価を行うこと。
• 達成率＝ΣW_1（実現機能数／要求仕様書技術機能数）

(4-3) 要求仕様設定手段の確認

(5) 知識ベースの保守性
曖昧性、不完全性を前提としたシステムにおいては、知識ベースの変更が頻繁に発生することを前提とする必要がある。従って、容易に知識ベースの変更かできる構造とすると共に、変更に伴って知識ベース全体の整合性が崩れないような保守ツールの整備が重要となる。

(5-1) 変更箇所発見の容易性評価
(C04)トレースできる機能があること。
説明等のために推論のトレースが明確なステップで行える必要がある。

(5-2) 知識ベースの理解性評価
(C03)知識の階層構造などが明確になっていること。
頻繁に知識が追加／変更されるために明確化が必要てある。
(C04)知識の可読性が高いこと。
メタ知識も容易に理解できるようになっているか。
(*M1)可読性の定量的な評価を実施すること。
• 可読知識数／全知識数

(5-3) 変更箇所の局所性評価
(*C1)知識の変更による副作用等の影響伝播が確認できること。
知識の追加により、他の知識への影響を評価できる仕組みが必要てある。

(5-4) 知識ベースの統一性評価

(5-5) 品質安定性の確認方法の確認

(5-6) 保守マニュアルの整備状況の確認

(5-7) 保守用ツールの整備状況の確認
(C01)保守用環境は整備されていること。
頻繁に変更があるため高機能な知識エディタ等が必要である。複数の知識ベースを同時に参照したり、変更する知識エディタが望ましい。

(6) ヒューマンインタフェースの妥当性
知的ＣＡＩシステムでは、知識処理部はオーサリング（教材呈示）システムに組み込まれる形となるので利用者インタフェースは表に現れてこない。しかし、教授戦略に基づいた最適の教材を呈示する必要があるので、知識処理部の出す結

果と呈示する教材の幅と深みとは密接に関連する。従って、知的ＣＡＩシステムでは、教材作成時から推論システムとの連携を考慮した利用者インタフェースの開発を行う必要がある。

構築時、保守時のインタフェースは、特に頻繁に変更作業が発生することを考慮した変更の容易性を実現したものとする必要がある。また、知識ベースの完備性、一貫性を容易に検証できる知識構造の図的表示機能などを具備したインタフェース機能が望ましい。

(6-1) 利用時のインタフェース評価

(*C1)既存システムとのインタフェースが考慮されていること。
　　既存のオーサリングシステムとのインタフェースが必要である。

(C13 ～C19)説明機能が充実していること。
　　学習者が納得のいくまて，きちんとした説明ができること。

(C02)操作性がよいこと。
　　人間工学的評価がなされている必要がある。

(C06)利用者の習熟度に応じたガイダンスであること。
　　学習者のレベルにあった教示結果に対する説明機能が必要てある。

(C04)マニュアルがなくても操作が可能なこと。
　　学習者は、操作のための勉強はしない。
　　操作者はシステムに関して全くの素人てあるので、誤操作に対する対策が必要となる。

(*C2)利用者にとって楽しいインタフェースの工夫がなされていること。
　　学習者の理解を促進するような工夫がなされているか。

(6-2) 保守時のインタフェース評価

(C01)知識ベース開発者用のインタフェースが考慮されていること。
- 複合的な知識を網羅的に表示する機能はあるか。
- 解析部と計画部に渡った推論経過を連続的に表示する機能はあるか。

(7) システムの総合評価

複合型システムでは、解釈部と計画部が連携して問題を解決する必要かあるので、全体の整合性を考慮した評価法が重要となる。知的ＣＡＩシステムで最も難しい問題が未知事象に対する評価法となる。浅い知識の表現ては、知識の網羅性の保証が非常に難しく、生徒の誤り原因が推定不可能な状態が発生するてあろう。その場合の対処法として、知識ベース構築者に不足の知識を促す（ロク等て）ための機能を考慮する必要がある。つまり、知的ＣＡＩのような非常に難しい対象を知識化するシステムの総合評価としては、運用時における未知事象にとのよう対象できるかという点まて考慮したシステムであるか否かが重要なポイントとなる。

(7-1) テーマ選定の妥当性評価

(C02)知識やノウハウの収拾が可能な問題であること。
　　充分に誤りデータが収拾できる環境、あるいは対象問題てあるか。

(*C1)モデル化が可能な問題であること。
　　誤りデータに基づく誤答のモデル化が可能な問題てあるか。

(*C2)利用効果が把握できる問題であること。
- 学習効果が定量的に把握できるような対象問題か。
- 学習者の学習意欲を向上させるようなシステムか。
- 学習者が発見的に誤りを学習できるような仕組みがあるか。

(7-2) 利用者の満足度評価
(C06)操作が簡単であること。
学習者が一人で学習できるか。
(C09)必要最小限の入力で済むようになっていること。
うんざりするような質問が出ないように工夫されているか。
(C12)説明機能があり，分かりやすいこと。
自己学習に耐えられるシステムであるか。

(7-3) 統合性の評価
(C02)他システムとの連続性があること。
特に，オーサリングツールとの連携は必須である。

(7-4) 成長可能性の評価
(C06)開発環境は，実用化レベルへの発展を考慮にいれること。
学習者用の実行マシンは，別マシンであり，成長性を事前に考慮する。

(7-5) 開発／保守用トキュメントの評価

(7-6) 開発方法の適切さの確認

(7-7) ツール・言語選択の適切さの確認

(7-8) システム開発の生産性評価

(7-9) システムのコストパフォーマンス評価

(7-10)セキュリティ対策の評価

《参考文献》
[Kobayashi 1988] 小林重信：知識工学、昭晃堂、昭和６１年.

3.6 評価に用いられるテスト方法

3 6.1 概　要

ここでは、エキスパートシステムの検証および妥当性検査（V&V Verification and Validation）を行う際に使用可能で有用であると思われるテスト方法（確認手段）を網羅的に分類して紹介する。

評価に際しては、評価対象のエキスパートシステムの重大度（リスク、安全性の面）・特質や開発・評価コストなどを勘案して、ここで紹介されているテスト方法を適宜選択し、テストをする必要がある。

以下で紹介するテスト方法の分類体系・次元は半入れ子構造になっているので、明確にするために分類番号を付与してある。

なお、ここに紹介されているテスト方法に関して、必ずしも、すべてのテスト方法を任意に組み合わせて選択して、エキスパートシステムのテストに使用出来るわけではなく、この半入れ子構造を意識して、テスト方法を適宜選択して、組み合わせて使用する必要がある。

3.6.2 エキスパートシステムのテスト方法の分類体系・次元

エキスパートシステムのテスト方法は、次に示すようにな次元て分類される。

次元X：妥当性を検査する方法による分類
- X-1：内容的妥当性テスト
- X-2：基準に基づく妥当性テスト
- X-3：構成概念的妥当性テスト
- X-4：表面的妥当性テスト
- X-5：未来予測的妥当性テスト
- X-6：併存的妥当性テスト
- X-7：過去予測的妥当性テスト

次元Y／Z：テスト対象を実行するか否かによるテスト方法の分類
- Y　：静的テスト／分析
- Z　：動的テスト

次元Y：静的テスト／分析でのテスト手段による分類
- Y-1：人手による
  - Y-1-1：レビュー
  - Y-1-2：ウォークスルー
  - Y-1-3：インスペクション
  - Y-1-4：変則評価
  - Y-1-5：リスク分析
  - Y-1-6：ハザート解析
  - Y-1-7：故障の木解析（FTA）
- Y-2：静的分析ツールによる
  - Y-2-1：ルールベースの論理的一貫性分析
  - Y-2-2：ルールベースの論理的完備性分析
  - Y-2-3：ルール間の相互参照分析
  - Y-2-4：確信値の矛盾検出

Y-2-5：データフロー分析
Y-2-6．制御フロー分析

次元Z：動的テストでのテストサイクルなどによる分類（図3 6-1参照）
Z-1：テストデータ（問題）の選定方法による分類
Z-2：テスト対象の内部構造を見て、その結果に基つくか、あるいは外部から見た機能に基つくかでのテスト方法の分類
Z-3：同しテストデータを再度使用してテストをするか否かによるテスト方法の分類
Z-4：テストを行う場所でのテスト方法の分類
Z-5：特定の目的に特化したテストでのテスト方法の分類
Z-6：推論過程の正しさを調べる手段による分類
Z-7：テスト結果の比較対象／手段による分類
Z-8：評価の正しさを評価するための手段による分類
Z-9：評価者の偏向を取り除く手段による分類

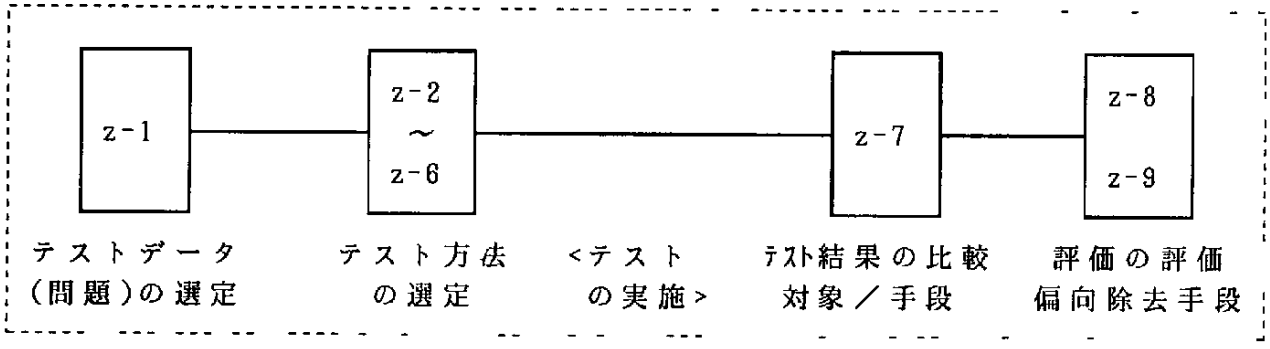


図3 6-1 動的テストのテストサイクル

次元Z-1：テストデータ（問題）の選定方法による分類
Z-1-1：ランダムデータ法
Z-1-2：同値分割法
Z-1-3：限界値分析法
Z-1-4：原因ー結果グラフ法
Z-1-5：層化抽出法
Z-1-6：誤り推定法
Z-1-7：運用プロフィール法
Z-1-8：実データ法（本番データ法）
Z-1-9：標準テスト問題法（能力レベル別）
Z-1-a：テスト網羅計測法（テストデータの質の評価法）
Z-1-b：変異テスト法（テストデータの質の評価法）

次元Z-2：テスト対象の内部構造を見て、その結果に基つくか、あるいは外部から見た機能に基づくかでのテスト方法の分類
Z-2-1：ホワイトボックステスト／構造テスト
Z-2-2：ブラックボックステスト／機能テスト

次元Z-3：同じテストデータを再度使用してテストをするか否かによるテスト方法の分類
Z-3-1：回帰テスト

Z-3-2：新規テスト
次元Z-4：テストを行う場所でのテスト方法の分類
Z-4-1：フィールドテスト
Z-4-2：実験室テスト
次元Z-5：特定の目的に特化したテストでのテスト方法の分類
Z-5-1：負荷テスト
Z-5-2：感度分析テスト
Z-5-3：削除テスト
次元Z-6：推論過程の正しさを調べる手段による分類
Z-6-1：説明機能
Z-6-2：プロトコル分析（内観分析）
次元Z-7：テスト結果の比較対象／手段による分類
Z-7-1：過去／既存のデータ／結果
Z-7-2：１人の専門家
Z-7-3：専門家のパネル（複数の専門家）
Z-7-4：対照基準グループ
Z-7-5：併行運用
Z-7-6：シミュレーション
Z-7-7：他のモデル
Z-7-8：実　験
Z-7-9：実地調査
次元Z-8：評価の正しさを評価するための手段による分類
Z-8-1：専門家間の一致係数
Z-8-2：信頼区間
Z-8-3：第Ⅰ種の誤り
Z-8-4：第Ⅱ種の誤り
次元Z-9：評価者の偏向を取り除く手段による分類
Z-9-1：チューリングテスト／目かくし評価
Z-9-2：クロスチェック
Z-9-3：統計的手段

以下では、この分類体系・次元に従って、テスト関連用語を解説することにする。なお、以下では、<x-x>は3 6.2項の対応する分類番号を示している。

3.6 3 妥当性を検査するテスト方法

エキスパートシステムの能力（performance or competency）の評価は人間の能力の評価とかなり類似した点も多いので、エキスパートシステムの妥当性テストでは妥当性（validity）に関して、心理学の分野で発展したテスト理論やテストそのものの妥当性を調べる概念や、行動科学での研究法を援用している。そのため、用語のアナロジィが行われおり、その語源そのものを正しく把握していないと、その用語が不自然に感じられるので、ここでは心理学分野での定義も併記しておく。なお、心理学用語の定義では、[Ikeda 1973]、[Takeuchi et al 1989]を参考にした。

妥当性を検査する代表的なテスト方法としては、下記に示す手法がある。

①内容的妥当性テスト<X-1>
②基準に基づく妥当性テスト<X-2>
③構成概念的妥当性テスト<X-3>
④表面的妥当性テスト<X-4>

基準に基づく妥当性テストは下記に示すようにさらに展開される。

⑤未来予測的妥当性テスト<X-5>
⑥併存的妥当性テスト<X-6>
⑦過去予測的妥当性テスト<X-7>

(1)内容的妥当性テスト<X-1>

【心理学】内容的妥当性(Content validity)とは、テストに用いられる課題内容(content)(テスト問題)がそれを用いて結論しようとしている測定内容のいかによい見本(サンプル)となっているかを示す指標である。例えば、授業で教えている教科内容がテストの内容にどの程度反映されているかであり、教科的妥当性(Curricular validity)とも呼ばれる。

【知識工学】内容的妥当性テスト<x-1>とは、システムが何を知っているか、何を知らないか、あるいは何を不正確に知っているかを確認することである[O'Leary 1987]。このテストでは、例えば、①専門家によるシステムの直接的な精査、②専門家によるチューリングテスト<z-9-1>、③他のモデル<z-7-7>との比較、などが行われる[O'Leary 1987]。

(2)基準に基づく妥当性テスト<X-2>

【心理学】基準に基づく妥当性(Criterion-related validity)がテストの妥当性を調べる際に最も広く利用されおり、その妥当性は予測的妥当性(Predictive validity)と併存的妥当性(Concurrent validity)に細分される。予測的妥当性では、例えば、適性検査てのテストの得点が、その人のその後に予定されていた作業での成績と高い相関があれば、その適性検査は予定された作業に対して高い予測的妥当性があるという。この場合の基準(criterion)は、その後の作業での成績である。一方、併存的妥当性では、例えば、精密な個人知能検査Aの代わりに、短時間で多数の人を同時に行う集団知能検査Bを実施し、AとBのテストの得点の間に高い相関があれば、そのテストBはテストAに対して高い併存的妥当性があるという。この場合の基準は、テストAの得点である。

【知識工学】基準に基づく妥当性テスト<x-2>では、システムの妥当性を確かめるために、例えば、エキスパートシステムの専門知識のレベルを確かめるために、基準を使用する[O'Leary 1987]。その基準としては、例えば、エキスパートシステムの専門知識に対しては専門家が保持している専門知識、知識ベースに対しては一貫性、正確性、完備性が採用されている[O'Leary 1987, O'Keefe & O'Leary 1989]。

未来予測的妥当性テスト(Predictive validity test)<x-5>では、未来のある時点で比較すべき基準データが入手できる状況を仮定している。同しテスト問題に対して、エキスパートシステムが導出した結果と、未来に起こった現実の結果・観測値とを比較して、エキスパートシステムの能力が評価される[Benbasat & Dhaliwal 1989]。併存的妥当性テスト(Concurrent validity test)<x-6>では、同時に同じテスト問題を解いてくれる専門家あるいは既存システムがあることを

仮定している。同じテスト問題に対して、エキスパートシステムが導出した結果と、専門家あるいは既存システムが出した結果とを比較して、エキスパートンステムの能力が評価される[Benbasat & Dhaliwal 1989]。そして、過去予測的妥当性テスト（Post-dictive validity test）<x-7>では、過去に入手した、あるいは既存・既知の結果データがあることを仮定している。同しテスト問題に対して、エキスパートシステムが導出した結果と、既知の結果データとを比較して、エキスパートシステムの能力が評価される[Benbasat & Dhaliwal 1989]。

（３）構成概念的妥当性テスト<X-3>

【心理学】構成概念的妥当性（Construct validity）とは、諸構成概念（construct）（心理学的構成体、例えば、不安）、あるいは因子（factor）を測定するように工夫された、例えば２つのテストＡとＢがあった場合、とちらのテストが対象とした１つの構成概念をより正しく測定しているかの指標てある。その際には、測定しようとしている各構成概念を厳密に定義し、かつ、正しいテスト結果が何であるかを明かにしておかなければならない。構成概念的妥当性は因子的妥当性（Factorial validity）と呼ばれることもある。

【知識工学】構成概念的妥当性テスト<x-3>では、エキスパートシステムが拠り所としているその対象分野の基本理論（原理、原則）があることを重要視している[O'Leary 1987]。このテストでは、①エキスパートシステムと第一原理からのそれぞれの結果、②第一原理と専門家からのそれぞれの結果、③エキスパートンステムと専門家からのそれぞれの結果との比較などが行われる[O'Leary 1987, O'Keefe & O'Leary 1989]。

（４）表面的妥当性テスト<x-4>

【心理学】表面的妥当性（Face validity）は、厳密な科学的手段を使用して確かめられたテストに対する妥当性ではなく、むしろ直観的・印象的な感しをもとにして得られたテストに対する妥当性の指標である。

【知識工学】表面的妥当性テスト<x-4>は、プロジェクトチーム要員、ユーザ、専門家が、自分の知識と直観を使用して、システムの問題解決能力と専門家のそれとを主観的に比較するアプローチである[O'Leary 1987, O'Keefe et al 1987]。このテストの目的は、タイムリーに、コスト有効性の高いやり方て、エキスパートシステムに対するシステム設計者、ユーザ、専門家の３者のそれぞれの視座の間の一貫性を検証することである[O'Leary,T et al 1990, Wright 1990]。

以上、妥当性に関する諸概念を紹介したが、これらの概念はいずれも他と明確に区別される独立の概念ではなく、相互に重複した部分もかなりあり、妥当性に関する評価はあらゆる角度から総合的に行われなければならない[Ikeda 1973]。

３６４ 静的テスト／分析および動的テストの概要

エキスパートシステムの品質を向上させるには、エキスパートンステムはソフトウェアの一種であり、ソフトウェアはエンジニアリングンステムの一種てもあるので、これらの分野で高度・大規模ンステムに対して有効性が実証された開発・検証・管理技術の根本思想（原理、原則）を積極的に活用することが要締てある。即ち、エキスパートンステムのテストに関しても、ソフトウェアに対するテ

ストや、エンジニアリングシステムに対するテストの考え方か十分適用可能て、有効である。

テストはテスト対象を実行させるか否かによって、次の２つに大別される。

①静的テスト／分析<Y> …… テスト対象を実行させない

②動的テスト<Z> …… テスト対象を実行させる

また、テストをする手段が人手を主体としたものか、コンピュータを活用したツールを主体としたものかによって類型化することもてきる。人手を主体とした手法は、レビューとか、ウォークスルーとか、インスペクションとか呼ばれる。ソフトウェアでは、最近は、重大な欠陥の検出能力の点で動的テストに比へてレビューの方が優れていることが実証されたこともあり、レヒューを重視するようになってきている。

テストに際して、テスト対象に対する分析や解析を主眼にしている場合には、テストという用語よりも、分析や解析という用語（英語てはanalysis）がしばしば用いられる。

動的テストは、テスト対象をブラックボックスとして扱うか、ホワイトホックスとして扱うかによって、更に次の２つに大別される。

①ブラックボックステスト<Z-2-1>

②ホワイトボックステスト<Z-2-2>

ブラックボックステスト<Z-2-1>は、主として要求仕様書（特に機能要件）に規定された機能が正しく実現されているかを調へるために行われるので、機能テスト<Z-2-1>とも呼ばれる。一方、ホワイトボックステスト<Z-2-2>は、システムの内部構造が設計仕様書に規定された通りに正しく実現されているかを調へるために行われるので、構造テスト<Z-2-2>とも呼ばれる。

3.6 5 静的テスト／分析

静的テスト／分析は、その手段によって、更に次の２つに大別される。

①人手による手法<Y-1>

②自動化ツールによる静的テスト／分析<Y-2>

3.6.5.1 人手による手法<Y-1>

人手による代表的な方法としては、下記に示す手法がある[Deutsch & Willis 1988]。

①レビュー<Y-1-1>

②ウォークスルー<Y-1-2>

③インスペクション<Y-1-3>

④変則評価<Y-1-4>

(１)レビュー<Y-1-1>

レビューとは、「人手によってトキュメントやプログラムが正しいかを克明に調べる」ことであり、特に、デザインレビューは、「アイテムの設計段階で、性能・機能・信頼性などを考慮しながら設計について審査し改善を図ること。審査には、設計・製品・検査・運用などの各分野の専門家が参加する」（JIS用語）であり、「製品の設計品質およびそれを具現化するために計画された製造・輸送・据付け・使用・保全なとのプロセスについて、客観的に知識を集めて評価し、改

善点を提案し、次の段階に進みうる段階にあることを確認する組織的活動の体系」ということができる[Miura et al 1985]。

エキスパートシステムのデザインレビューにおいて、今後重要となる分析／解析は、リスク分析（Risk analysis）とハザード解析（Hazard analysis）である[Leveson 1986]。

リスク分析<Y-1-5>は、「システムの誤りによって起きると考えられる損失を予測するためのプロセス」と定義される。分析の目的は、リスクにさらされる可能性を確認し、そのリスクの程度を数量的に推測し、そのリスクを優先順位別にランクし、そして、コスト効果の高い解決を可能にすることである。

エキスパートシステムが医療・原子力・航空・宇宙・防衛・化学工業のような安全性が致命的な分野で、リアルタイムで直接的な監視・制御に使用された場合、それ自身および全体システムの安全性が重大な問題となってくる。エキスパートシステムの安全性が実証されないかぎり、これらの分野でのエキスパートシステムの利用は限定されるであろう。（特に、エンジニアリングにおける有名なMurphyの法則*を常に意識することが肝要である[Bell 1989]。）

［注］*：Murphyの法則は一言でいえば、複雑なエンジニアリングシステムにおいて、極端に確率が小さいが、悪いことが起こる可能性があれば、確率とは関係なく、使用中に、その悪いことが必ず起こるという法則である。

信頼性（Reliability）は、一般に、システムが特定の諸環境条件下で、意図した機能を遂行する確率と定義されている。一方、安全性（Safety）は、意図した機能が遂行されているか否かにかかわらず、事故（mishap）を引き起こす諸条件が生じない確率と定義されている。しばしば、信頼性と安全性は同義であるとみなされるが、この２つの概念を明確に区別することが、その実現・対策を、あるいは、評価を考える上で、要締である。

信頼性での故障（Fault）に対して、安全性では、人間または資材に損害を与える実在あるいは潜在の状態を“ハザード”（Hazard）と呼ぶ。ハザード解析<Y-1-6>は、システムにおけるハザードを識別し、その影響のきびしさと、発生確率を調べ、安全性を評価する方法である。

ソフトウェアのハザード解析<Y-1-6>では、ソフトウェアのハザードと、安全性に重大な影響を及ぼす単一および複合の障害（failure）のシーケンスを識別し、タイミング要件を含めたソフトウェア安全性要件を決定し、ソフトウェアの安全性を解析して、計測する。ソフトウェア安全性要件には、システムが何をなすべきかだけでなく、何をしてはならないかも含まれていなければならない。

ハザード解析の補助手段である故障の木解析（FTA　Fault Tree Analysis）<Y-1-7>では、木の頂上（根）にシステムの使命達成上、好ましくない事象（故障）をあげ、次にその発生の原因となると思われるすべての事象をあげ、これらをANDあるいはORなどの論理記号で結び、さらに、それらの事象の原因となる副事象を調べ、順次木の枝のように組み立てていき、故障発生の経路とその根本原因をつきとめる解析方法である（図3 6-2参照）[Leveson 1986]。

ソフトウェアのＦＴＡでは、ソフトウェアの詳細設計あるいはコーディングの段階が終了した時点で、ＦＴＡ手順を使用して、故障の木の頂上にあるクリティカルな制御故障から後向きに遡行して、プログラムが故障の木の頂上の故障（事故）を引き起こす原因となるかどうか検証する。

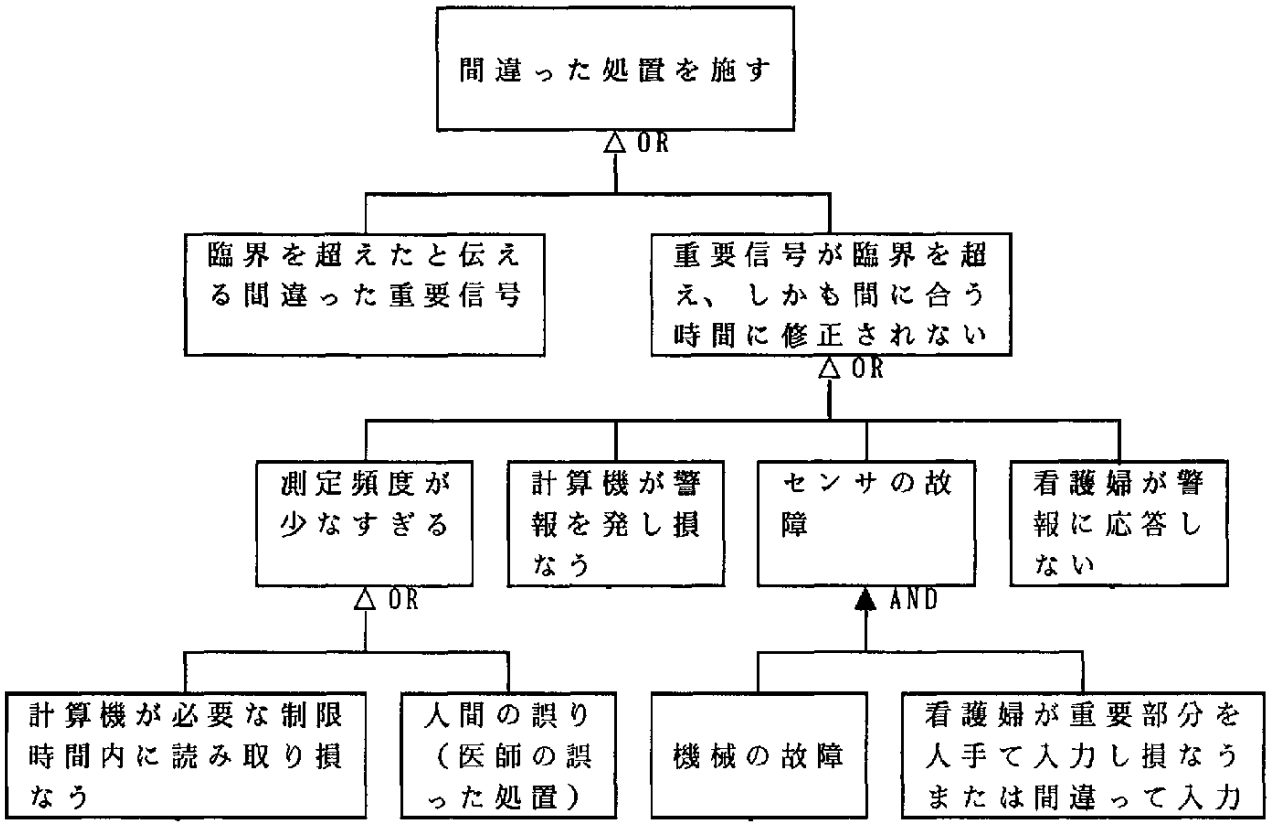


図3.6-2　故障の木解析（ＦＴＡ）の例：患者監視システムの場合

デザインレビューを実施することにより、エキスパートシステム開発の工程短縮、品質およびコスト有効性（Cost effectiveness）の評価および確保が可能となる。

（２）ウォークスルー<Y-1-2>

ウォークスルー（Walk-through）もレビューの一種て、「１人の設計者またはプログラマが開発チーム内の他の要員を指導して、彼らが書いた設計書またはコートのセグメントを通して、技法、仕事のやり方、犯しやすい誤り、開発規格に対する違反やその他の問題について他の要員に質問したり、意見を述へたりする再検討の過程」（ANSI／IEEE用語）[ANSI/IEEE 1983]てあり、特に、コートウォークスルーは、「人手でプログラムコートの１行１行が正しいかを克明に調べる」ことであり、知識ベースや問題解決・推論プログラムを１行１行正しいかを克明に調へることにより、信頼性を向上させることが可能となる。

（３）インスペクション<Y-1-3>

インスペクション（Inspection）もまたレビューの一種で、「ソフトウェア要求仕様書、設計書またはコートを、作成者以外の個人またはグループが詳細に調査し、欠陥や開発標準との不適合およびその他の諸問題を発見するための公式の評価手法」（ANSI／IEEE用語）[ANSI/IEEE 1983]である。インスペクションはウォークスルーとにているが、ウォークスルーに比へてより公式的な色彩が強い評価手法である[Deutsh & Willis 1988]。

(4)変則評価<Y-1-4>

変則(Anomaly)とは、「ソフトウェアの文書内や動作中に見つけられた欠陥で、以前に検証されたソフトウェア製品あるいは参考した文書に基づく予想とかけはなれたもの」である。変則評価は予想とかけはなれた欠陥を発見する評価手法である[Levinson & Daughtrey 1990]。

3 6 5.1 静的テスト/分析ツール<Y-2>

静的テスト/分析ツールによる代表的なテスト方法としては、下記に示す手法がある。

①ルールベースの論理的一貫性分析<Y-2-1>
②ルールベースの論理的完備性分析<Y-2-2>
③ルール間の相互参照分析<Y-2-3>
④確信値の矛盾検出<Y-2-4>
⑤データフロー分析<Y-2-5>
⑥制御フロー分析<Y-2-6>

静的テスト/分析を自動化するツールとして、まず最初に、知識ベース、特にルールベースを対象とするものが研究開発されているので、紹介する[O'Leary & O'Keefe 1989]。次に、従来のソフトウェアのテストで使用されている静的分析もエキスパートシステムに適用可能であるので、それについても述べる。

(1)ルールベースの論理的一貫性および完備性の分析<Y-2-1>、<Y-2-2>[Nguyen et al 1987]

ルールベースの論理的一貫性を阻害するルールは、次に示すように分類される<Y-2-1>。

- 冗長ルール(Redundant rule)
- 競合ルール(Conflicting rule)
- 包摂ルール(Subsumed rule)
- 不必要なIFの条件項を持つルール(Unnecessary IF Conditions)
- 循環ルール(Circular rule)

そして、ルールベースの論理的完備性を阻害する誤りは、次に示すように分類される<Y-2-2>。

- 未参照の属性値(Unreferenced attribute values)
- 誤った属性値(Illegal attribute values)
- 欠如ルール(Missing rule)
- 未到着な結論部を持つルール(Unreachable conclusions)(IFの条件部が決して発火しない)(後向き推論の場合)
- 袋小路のIFの条件項を持つルール(当該IFの条件項と同し結論項を持つルールや事実がない)/袋小路の目標(目標と同し結論項を持つルールがない)(Dead-end IF conditions and dead-end goals)(後向き推論の場合)

これらの論理的な一貫性およびの完備性を阻害するルールの検出は、現在自動的に検出することが可能となってきており、一部商用ノール化されている。

(2)ルール間の相互参照リスト<Y-2-3>

ルール間の相互参照リスト、即ち、ルールのTHEN部の条件と同し条件をIF部に

持つルールとの相互参照のマトリックスの作成とか、IF部にある変数の条件式が、変数の属性や定義域からみて妥当であるかなどを自動的にチェックすることも可能である[Rossomando 1989]。これらのツール機能は、ルール間の相互作用をレビューする際に有用である。

(3)確信値の矛盾検出<Y-2-4>

ルールに確信値S(Sufficency factor)とN(Necessity factor)を付けるBayesian的な方式では、そのSとNの間の論理的矛盾を自動的に検出することも可能になってきている[O'Keefe & O'Leary 1989]。

(4)データフロー分析<Y-2-5>

データフロー分析は、プログラム中の変数に対する値の定義とその変数の相互参照関係を解析する方法である[Tamai et al 1988]。この分析は、<Y-2-1>や<Y-2-2>と類似した方法であり、この分析を行うことにより、知識ベース中の不必要な知識を発見可能となる。

(5)制御フロー分析<Y-2-6>

制御フロー分析は、プログラムを実行せずに、ソースコードを調べて、実行の流れ(フロー)に関する情報を収集する方法である[Tamai et al 1988]。この分析を行うことにより、ルールの発火順序や推論過程を精査可能となる。

3.6.6 動的テスト

動的テストでは、下記に示すテストサイクルなどの各項目を考慮する必要がある。

①テストデータ(問題)の選定方法<Z-1>
②テスト対象の内部構造を見て、その結果に基つくか、あるいは外部から見た機能に基つくかでのテスト方法<Z-2>
③同じテストデータを再度使用してテストをするかでのテスト方法<Z-3>
④テストを行う場所でのテスト方法<Z-4>
⑤特定の目的に特化したテストでのテスト方法<Z-5>
⑥推論過程の正しさを調べる手段<Z-6>
⑦テスト結果の比較対象/手段<Z-7>
⑧評価の正しさを評価するための手段<Z-8>
⑨評価者の偏向を取り除く手段<Z-9>

3 6 6 1 テストデータ(問題)の選定方法

エキスパートシステムを効率的に、効果的にテスト・評価するには、よいテストデータ(問題)を選定することが要締である。テストデータ(問題)の代表的な選定方法としては、下記に示す方法がある[Pressman 1982, Ishii 1986, Tamai et al 1988, Tsai & Zualkerman 1990]。

①ランダムデータ法<Z-1-1>
②同値分割法<Z-1-2>
③限界値分析法<Z-1-3>
④原因ー結果グラフ法<Z-1-4>

⑤層化抽出法<Z-1-5>
⑥誤り推定法<Z-1-6>
⑦運用プロフィール法<Z-1-7>
⑧実データ法（本番データ法）<Z-1-8>
⑨標準テスト問題法（能力レベル別）<Z-1-9>
⑩テスト網羅計測法（テストデータの質の評価法）<Z-1-a>
⑪変異テスト法（テストデータの質の評価法）<Z-1-b>

（１）ランダムデータ法<Z-1-1>
ランダムデータ法は、入力データ（問題）の母集団（入力データ領域）から無作為に取り出して、それらをテストデータ（問題）とする方法である。この方法では、テストでの目標とする網羅率を保証するには、たくさんのテストデータを選択する必要がある。

（２）同値分割法<Z-1-2>
同値分割法（Equivalence partitioning）は、プログラムの入力データ領域を同じ出力結果になるものに分割し、その分割した副領域から代表的な値を取り出して、それらをテストデータ（問題）とする方法である。この方法を実行する際に、入力データ領域を分割するための基準が明確でないことがある。

（３）限界値分析法<Z-1-3>
限界値分析法（Baundary value analysis）は、境界条件となる値を試すようなテストケースを取り出して、それらをテストデータ（問題）とする方法である。ソフトウェア誤りの多くは、インデックス、データ構造、スカラーなとの値のとる範囲での境界の直前、その値、直後で起こっているので、この方法は非常に有効である。この方法を実行する際に、限界や境界の定義が明確てないことがある。

（４）原因ー結果グラフ法<Z-1-4>
原因ー結果グラフ法（Cause-effect graph）は、テスト対象とするプログラムの仕様を入力条件、環境条件などの原因と、処理、出力なとの結果との論理的関係をグラフに表現し、このグラフから決定表（Decision table）を作成し、そしてテストケースを取り出して、それらをテストデータ（問題）とする方法てある。この方法では、システム全体をテストするには、可能な組合せの個数が膨大となってしまう。

（５）層化抽出法<Z-1-5>
層化抽出法（Stratified sampling）は、入力データ（問題）の母集団をいくつかの部分母集団に分割し、各部分母集団からサンプルデータを取り出して、それらをテストデータ（問題）とする方法である[O'Keefe et al 1987, Takeuchi et al 1989]。この部分母集団を層（stratum）といい、分割操作を層化（層別）という。層化の際に層内がなるへく均質（例えば、ほぼ同し難易度のテストデータ（問題））になるようにすると、テストデータの有効性が向上する。この方法は、同値分割法<Z-1-2>と考え方が類似している。

(6)誤り推定法<Z-1-6>

誤り推定法(Error guessing)は、同種のアプリケーションに対するプログラマの直観と経験に基づいて、誤りやすい個所を標的としたテストケースを取り出して、それらをテストデータ(問題)とする方法である[Tsai & Zualkerman 1990]。この方法の効力は、テスト者の経験や能力に依存する。

(7)運用プロフィール法<Z-1-7>

運用プロフィール法(Operational profile)は、実際の運用において、ユーザがソフトウェアをどう使っているかを分析して、そのデータケースの使用頻度を反映してテストケースを取り出して、それらをテストデータ(問題)とする方法である[Musa 1989]。

(8)実データ法(本番データ法)<Z-1-8>

実データ法(本番データ法)は、テストデータ(問題)として本番データを使用する、あるいは、本番使用予定のデータをそのまま流用して、テストデータ(問題)とする方法である。この方法は、セキュリティ上、問題となるであろう。

(9)標準テスト問題法(能力レベル別)<Z-1-9>

標準テスト問題法は、その専門領域の学会や業界で広く認められている能力別の標準的な試験問題を使う方法である[Miller 1989]。これは、等級別の資格認定・検定試験の試験問題と同じ考え方である。

(10)テスト網羅計測法(テストデータの質の評価法)<Z-1-a>

テスト網羅計測法(Test coverage)は、機能仕様に基づいて作成したテストデータを用いて、プログラムを実行した際に、それらのデータによって、例えば、プログラムの全体の何%のステートメントが実行されたかを計測することによって、テストデータの質を評価する方法である[Tamai et al 1988]。その際に、テスト目標とした網羅率が達成されなかった場合には、テストデータをさらに追加して、テストの網羅率を高めることになる。

(11)変異テスト法(テストデータの質の評価法)<Z-1-b>

変異テスト法(Mutation test)は、プログラム中に存在する誤りの検出に対してのテストデータの有効性を確かめ、評価する方法である[Tamai et al 1988, Tsai & Zualkerman 1990]。このテストでは、元々のプログラムに対して系統的なやり方(disciplined way)て改悪し、その一連のテストデータを用いて改悪されたプログラムの改悪部分の誤りを検出できるかを調へて、その種の誤りに対するテストデータの誤り検出能力、即ち、質を評価する。もし、その種の誤りか検出できなかった場合には、元々のプログラムにもその種の誤りが内在している可能性もあるので、その種の誤りを検出できるテストデータをさらに追加して、テストを行う必要がある。

3 6.6 2 ホワイトボックステスト/構造テストとブラックボックステスト/機能テスト

テスト方法は、テスト対象の内部構造を見て、その結果に基づいてテストを行うか、あるいは外部から見た機能に基づいてテストを行うかて、下記に示すよう

に分類される。

①ホワイトボックステスト／構造テスト<Z-2-1>

②ブラックボックステスト／機能テスト<Z-2-2>

(1)ブラックボックステスト／機能テスト<Z-2-1>

ブラックボックステストあるいは機能テストでのテスト対象は、実現した機能、主としてエキスパートシステムから導出される結果や結論である。

本テストを実施するにおいてもっとも重要な点はテストテータの選定である。一般に、ブラックボックステストあるいは機能テストを完璧に行うには膨大なテストデータを必要とし、その結果、膨大なテスト時間（データ準備、マシン実行などに要する時間）を必要とするので、この点を事前に検討しておく必要がある。

テストデータの選定は従来のソフトウェアのテストの際の選定基準や考え方と同じである。但し、エキスパートシステムは一般にソフトウェアと異なり100%正解の答を出すとは限らないので、解の正解率や網羅率を算出するためのテストては、統計的に見て、例えば実務的に出現頻度が妥当なテストデータを選定する必要がある。

結果や結論に対する動的テストで検査・評価すへき品質特性の例を下記に示す[Walters & Nielsen 1988]。

①一貫性（Consistency）…… 知識ベースのどのルールが使われようと、あるいはどんな推論経路をとろうと、同じ入力データに対し、どの程度同し結論に到達するかを表す度合である.

②完備性（Completeness）…… エキスパート・システムか対象としている問題領域で正しいと考えられている結論すへてが、との程度導出されるかを表す度合である.

③健全性（Soundness）…… 導出された結論すへてが、との程度正しいかを表す度合である.

④精　度（Precision）…… 入力データがほんの少し変化した場合に、出力がどの程度変化するかを表す度合である.

⑤応答時間性（Response time）…… 要求された時間内てとの程度結論を導き出せるかを表す度合である.

⑥入力順序独立性（Sequence independence）…… データの入力順序を変えても、導き出される結論がどの程度変化しないかを表す度合てある.

⑦強固性（Robustness）…… 誤りのあるデータ、不正確なデータ、または、一部のデータが欠落していても、導出される結論がとの程度正しい結論てあるかを表す度合である.

(2)ホワイトボックステスト／構造テスト<Z-2-2>

ホワイトボックステストあるいは構造テストでのテスト対象は、実現した機能そのものよりも、主として知識ベースに格納されているルール、フレーム、メソッドなどの形式の個々の知識と、結論を導出した推論過程である[O'Leary & O'Keefe 1989]。

個々の知識（特にプロダクションルール）をテストするには、その知識が発火、あるいは使われる条件を明確にし、その条件にあったテストデータを用意する必要がある。従来のソフトウェアのテストの網羅率と同種の網羅率が考えられるが、その網羅率の有効性をも勘案して、網羅率テストを実施する必要がある。

エキスパートンステムでは、確信度を扱うこともあるので、ホワイトホックステストを行って、確信度の伝播過程を検査することも重要である。

結論を導出した推論過程の検査は、通常トレース機能や説明機能<Z-6-1>を用いて行われるが(、現在の技術水準や開発者の負担を考慮に入れると)、なぜその推論過程を通ったかの正当化説明は後向き推論ではある程度成功しているが、前向き推論では単なるトレース情報にしかすぎず、説明になっていない。なぜならば、前向き推論では、ルールの発火の順序は実質的にはプロクラミングの際に決められているので、なぜそのルールが選ばれたかの意味つけができないからである。いずれにしろ、どちらの推論方式でも、説明機能<Z-6-1>は、基本的にはエキスパートシステムの挙動を忠実に再現するだけであり、ルールベースンステムにおいては、ルールの連鎖を示すだけである(これは、浅い知識による浅い説明のみを実現しているといえる。)。もし、前向き推論で正当化説明を可能にするには、プログラムの検証で1つ1つのステートメントに対してプログラマが前もって記述したアサーション(assertion)が必要なように、例えば、各ルールに人間が前もって記述した正当化説明文を付けておかなければならず、コスト有効性の点で実施されないであろう。別のアプローチとして、ルールベースの他に、深い知識として、領域に関する機能・構造・挙動の3種の知識を用意すれば、何故そのルールがその状況で使われたかのルールの正当性を説明することが可能となる(これは、深い知識による深い説明といえる。)[Yamaguchi et al 1987]。しかしながら、深い知識のエキスパートンステム化に関しては、今後の課題でもある。

エキスパートシステム自身による正当化説明の理解性を向上させるには、推論過程の連鎖を説明用に再構成する能力が要求されるが、説明再構成機能はまたまだ研究課題であり、動的テストでは利用できないであろう。そのため、テスト時には、人間が、その推論連鎖を正当化の説明をしつつ、調べることになる。

3 6 6 3 回帰テスト

テスト方法は、同じテストデータを再度使用してテストをするか否かで、下記に示すように分類される。

①回帰テスト<Z-3-1>

②新規テスト<Z-3-2>

(1)回帰テスト<Z-3-1>

回帰テスト(Regression test)とは、「ンステムまたはシステム構成要素に対する変更の際に生じた欠陥を検出し、意図しない有害な影響を生していないことを、または、修正後のシステムまたはンステム構成要素が規定された要求仕様にあっていることを確認する目的で、以前のテストで使用したテストデータの全部あるいは一部を使用して再度テストを行うこと」である(ANSI/IEEE用語)[ANSI/IEEE 1983, Pressman 1982]。

(2)新規テスト<Z-3-2>

新規テストは、回帰テストと対比させるために分類上、単に設けただけの用語であり、過去のテストでは使用されていない新しいテストデータを使用して行うテストである。

3 6 6.4 フィールトテストと実験室テスト

エキスパートンステムのテストをどこの場所で行うかによって、下記に示すように分類される。

①フィールドテスト<Z-4-1>

②実験室テスト<Z-4-2>

(1)フィールトテスト<Z-4-1>

フィールトテスト(Field test)はエキスパートンステムをフィールトに設置して、能力の欠陥を見つけるために行うテストである[O'Keefe et al 1987]。このテストは、開発者の立場から見ると、①テスト作業の責任をユーサ側に委譲し、②テスト中にエキスパートンステムが受け入れ可能な能力範囲を達成すれば、ユーザは問題点を報告するのをやめるので、受け入れ可能な能力範囲を暗黙に得られる点に特徴がある。しかし、フィールトテストは非クリティカルなアプリケーションでしか採用することができない。

(2)実験室テスト<Z-4-2>

実験室テスト(Laboratory test)では、エキスパートンステムを実験室内てテストするので、エキスパートンステムと比較される専門家(評価者)に起因する測定誤差の要因を制御することが可能となる[O'Leary 1987]。

3 6.6 5 特殊目的のテスト

特殊目的の代表的なテストには、下記に示すテストがある。

①負荷テスト<Z-5-1>

②感度分析テスト<Z-5-2>

③削除テスト<Z-5-3>

(1)負荷テスト<Z-5-1>

負荷テスト(Stress test)は、短時間に設計で想定した最大性能の限界点付近に達する大量の負荷をンステムにかけて行うテストである[Ishii 1986, Green & Keyes 1987]。例えば、リアルタイムシステムに対して、全端末から同時にトランザクションを入力するテストである。この方法を実行する際に、との負荷をテストすれば有効であるかが明かてないことがある。

(2)感度分析テスト<Z-5-2>

感度分析テスト(Sensitivity analysis test)は、エキスパートンステムに対する入力データの値を系統的に変化させて、システムの能力の感度を調へる目的で行うテストである[O'Keefe et al 1987]。特に特異点や運用限界点付近で、入力データをほんの少し変化させて、採用しているアルゴリズムや試行錯誤手法がどの位敏感であるかを調べる必要がある[Green & Keyes 1987]。

(3)削除テスト<Z-5-3>

削除テスト(Ablation test)は、システム中の微細構成要素(construct)すへてが要求仕様を満たすために必要であることを確認する目的で、ンステム中の微細構成要素を適宜削除して行うテストである[Tsai & Zualkerman 1990]。もし、それでも正しく機能すれば、削除された微細構成要素の部分は余分て、不必要て

あるとみなされる。しかし、ノステムがスパゲティみたいて、構造化されてない場合、その部分が本当に不必要であるかを決定するのが困難であることが多い。

3 6 6 6 説明機能とプロトコル分析

エキスパートシステムが結果を導出した際の推論過程の正しさを調へる典型的な手段としては、下記に示すものがある。

①説明機能<Z-6-1>

②プロトコル分析（内観分析）<Z-6-2>

（1）説明機能<Z-6-1>

説明機能（Explanation facility）は、ユーザがエキスパートンステムが結果を導出した際の推論過程を調へるためにエキスパートシステム構築ツール／ノェルに用意されている機構である。説明機能としては、推論過程のトレース情報を単に表示するものから、それを構造化して表示するもの、そして、ユーザが推論過程の正当性を調へるのを容易にする再構成の能力を備えたものまてがある。この説明機能を使用することにより、間違った答につながる推論のステップや、間違った推論のステップにつながる知識ヘース中の事実が明かになるので、必要な修正の手がかりを得ることができる。

（2）プロトコル分析（内観分析）<Z-6-2>

プロトコル（内観、Protocol）とは、「自分自身の精神状態やその動きを内面的に観察すること」てある。プロトコル分析では、専門家（被験者）にその問題を実際に解いてもらい、その際にその解決過程で使用された推論戦略、各推論ステップの根拠、用いられた知識を話してもらって、推論過程を分析する[O'Keefe & O'Leary 1989]。この分析結果を基にして、説明機能<Z-6-1>を使用して、エキスパートシステムの推論過程が妥当であるかの比較評価が可能となる。

一般に、プロトコル分析では、被験者にシステムを使用しながら考えていることをひとりごとを言うように話してもらうので、対話型ノステムてのヒューマンインタフェースの評価にも使用可能である。

3 6 6 7 テスト結果の比較対象および比較手段

テスト結果に対する代表的な比較対象および比較手段としては、下記に示すものがある。

①過去／既存のデータ／結果<Z-7-1>

②1人の専門家<Z-7-2>

③専門家のパネル（複数の専門家）<Z-7-3>

④対照基準グループ<Z-7-4>

⑤併行運用<Z-7-5>

⑥ンミュレーション<Z-7-6>

⑦他のモデル<Z-7-7>

⑧実　験<Z-7-8>

⑨実地調査<Z-7-9>

(1)過去／既存のデータ／結果<Z-7-1>
エキスパートシステムの結果の評価での比較対象として、すてにある過去の、あるいは既存のデータあるいは結果を使用する。

(2)1人の専門家<Z-7-2>
エキスパートシステムの結果の評価での比較対象として、1人の専門家に同し問題を解いてもらったその答を使用する。その場合、その専門家が知識を提供した本人であることが多いので、その専門家のクローンとしてエキスパートンステムを構築する場合は問題がないが、そうでない場合には、客観性の問題が発生する可能性がある。

(3)専門家のパネル(複数の専門家)<Z-7-3>
エキスパートシステムの結果の評価での比較対象として、複数の専門家からなるパネル(Panel)を使用する。エキスパートシステムが対象とする問題に対して絶対的な正解が存在しない、あるいは不明の場合、専門家によって、その問題に対する正解(意見)が異なる際には、専門家間の一致性<Z-8-1>の問題が発生する。現実の問題として、専門家間での正解に対する合意をどうやて形成するかを事前に検討しておく必要がる。

(4)対照基準グループ<Z-7-4>
多くのエキスパートンステムでは、ユーザとシステムとが協力して問題を解いているので、システム単独を評価対象にしてもあまり意味がない[O'Keefe & O'Leary 1989]。この場合には、対照基準グループ(Control group)という方法論が使用可能である。ケースとして、①システムを使用して問題を解くグループと、②システムを使用しないで問題を解くグループを設ける。②のグループは対照基準グループと呼ばれる。妥当性検査では、システムを使用したグループの方が、システムを使用しない対照基準グループよりも作業の質や生産性の点て勝っているかを評価することになる。

(5)併行運用<Z-7-5>
エキスパートンステムの結果の評価での比較対象として、併行運用されている既存のシステム(場合によっては業務担当者)が出力した結果を使用する。

(6)シミュレーション<Z-7-6>
エキスパートシステムの結果の評価での比較対象として、同一問題を解くンミュレーションシステムが出力した結果を使用する。しかし、ンミュレーンョンて使用されているモデルそのものが完全であるかの問題があらたに発生する[O'Keefe & O'Leary 1989]。

(7)他のモデル<Z-7-7>
ある問題解決に対して、最適化モデル(OR)あるいは統計モデルのような異なる型のモデルが存在する場合がある[O'Keefe & O'Leary 1989]。例えば、天気予測エキスパートシステムと回帰モデルとを比較し、とちらの予測結果が精度の点でよいかを評価する。

(8)実　験<Z-7-8>

エキスパートシステムの結果の評価での比較対象として、同一機能を実現している実験を行って、その実験結果を使用する。

(9)実地調査<Z-7-9>

実地調査は、エキスパートンステムの結果が正しいかを、実際の現場に行って調査して決める手法である。例えば、電話網の故障診断エキスパートンステムの場合、エキスパートンステムが指摘した個所がほんとうに故障しているかを実地に調査することは可能である。

3 6.6 8 評価の正しさを評価するための手段

テスト結果に対する評価の正しさを評価するための典型的な手段としては、下記に示すものがある。

①専門家間の一致係数<Z-8-1>

②信頼区間<Z-8-2>

③第Ⅰ種の誤り<Z-8-3>

④第Ⅱ種の誤り<Z-8-4>

①と②の詳細に関しては、例えば、参考文献[Takeuchi et al 1989]を参照されたし。

(1)専門家間の一致係数<Z-8-1>

エキスパートシステムの結果と複数の専門家の結果を直接比較したり、チューリングテスト<Z-9-1>を使用して間接的に比較したりする場合、専門家間の一致係数(Consistency measure)(観測者間の信頼性(Interobserver reliability)とも呼ばれる。)が主要な考慮事項になる[O'Keefe et al 1987, O'Keefe & O'Leary 1989, Takeuchi et al 1989]。一致係数としてよく使われる級内相関係数が評価者間の信頼性を評価するために使用可能である。テストケースjに対する専門家iの判定をY_{ij}で表すと、専門家の信頼性に関するモデルは

$$Y_{ij} = \mu + E_i + \varepsilon_{ij}$$

となる。ここで、μはすべての専門家の平均得点、E_iは第i番目の専門家の影響度合(即ち、平均値μからの偏差)、そして、ε_{ij}は誤差項である。このモデルから相関係数を求め、検定することも可能である。Y_{ij}が連続変数ではなく、類別変数(Categorical variable; 名義尺度(Nominal scale)とも呼ばれる)(例えば、成績に対する優、良、不可)である場合には、信頼性を測定するために、カッパー統計量(Kappa statistic)(κ)を使用することができる。

(2)信頼区間<Z-8-2>

区間推定を行って求めた推定値の上限値と下限値を信頼限界といい、その信頼限界にはさまれた区間を信頼区間という。また、信頼限界の間に母数が入っている確率、いいかえれば推定を誤らない確率を信頼率あるいは信頼性という。信頼率($1-\alpha$)は通常95%にとる。

エキスパートシステムが単一の結果を導出する場合には、専門家の結果と単純明解に比較することができる。しかし、もしエキスパートシステムの結果がX_i(ベクトル値)で、専門家の結果もY_iの場合、それらの差は、

$$D_i = X_i - Y_i$$

となる。n個のテストケースの場合には、n個の観測された差は、D_1からD_nとなる。エキスパートシステムと専門家の間の差に対する信頼区間は、

$$d - t_{n-1,\alpha/2}\, Sd/n, \quad d + t_{n-1,\alpha/2}\, Sd/n$$

となる[O'Keefe et al 1987, O'Keefe & O'Leary 1989]。ここで、dは差の平均値、Sdは標準偏差値、$t_{n-1,\alpha/2}$は自由度nのt分布から得られる値である。1-αは信頼率であり、前もって与えられる。信頼区間の間にゼロが存在すれば、エキスパートシステムと専門家との間に有意な差はないとみなすことかてきる。

(3)第I種の誤り<Z-8-3>

第I種の誤りは、仮説H_0が正しいのに、H_0を捨てる誤りてある[Miura et al 1985]。通常、あわてものの誤りともいわれる。第I種の誤りは、エキスパートシステムの開発コストを必要以上に上昇させてしまう[O'Keefe et al 1987]。

(4)第II種の誤り<Z-8-4>

第II種の誤りは、仮説H_0が正しくないとき、H_0を捨てない誤りてある[Miura et al 1985]。通常、ぼんやりものの誤りともいわれる。第II種の誤りは、悲劇的な結末を生じる[O'Keefe et al 1987]。例えば、病気の診断を誤って行う医療診断エキスパートシステムては、誤った処置によって患者を苦しめてしまう。それ故に、クリティカルなアプリケーションにおいては、第II種の誤りを最小にしなければならない。

3 6 6 9 評価者の偏向を取り除く手段

テスト結果に対する評価の際の評価者の偏向を取り除くための典型的な手段としては、下記に示すものがある。

①チューリングテスト/目かくし評価<Z-9-1>
②クロスチェック<Z-9-2>
③統計的手段<Z-9-3>

(1)チューリングテスト/目かくし評価<Z-9-1>

チューリングテスト(Turing test)(あるいは目かくし評価(Blinded evaluation))は、コンピュータに"知能"があるかを判定するテストて、特に評価者のエキスパートンステムに対する偏向を除去する有力な手段てある[O'Keefe et al. 1987]。

MYCINでは、菌血症の評価と脳膜炎の評価が行われ、チューリングテストの重要性を如実に示した[Gaschnig et al. 1983]。最初の菌血症の評価ては、MYCINの能力を評価した専門医は、評価対象がコンピュータの出力結果てあることを事前に知っていた。彼らのコメントや批評の大半は、医療分野におけるコンピュータの正しい役割に対する彼ら自身の偏向を反映したものであった(例えば、「コンピュータでは、患者を診ていないので、何の病気であるかの適切な診断は下せるとは思えない」という指摘が示している。)。この評価結果を踏まえて、脳膜炎の評価では、100個のテストケース(症例)に対し、MYCINが下した診断結果および治療方法(recommendation)と、同じ症例を被験者9名(伝染病が専門の医学部局員から医学生まで)に依頼して得た診断結果および治療方法とを混せて、外部からはどちらのそれなのかを分からないようにして、公的機関の専門医8名に診

断結果および治療方法の妥当性について意見を求めた。このような形式のテストを行うことにより、評価者の潜在的な偏向を除去することがてき、MYCINの能力のレベルが医学部専門医から医学生のどのあたりに位置するものてあるかが分かった。

エキスパートシステムがチューリングテストに合格することは必要条件ではあるが、このチューリングテストに合格したからといって、真の知能が実現されたかは別問題である。

(2)クロスチェック<Z-9-2>

クロスチェック(Cross-checking)は、各評価者の評価結果を相互に交換し、お互いの評価結果の妥当性を点検するやり方である[O'Keefe et al 1987]。

(3)統計的手段<Z-9-3>

テスト結果に対する評価の際の評価者の偏向を取り除くための統計的手段としては、例えば、下記に示すものがある[O'Keefe et al 1987]。

①t検定(2標本問題)(Paired t-tests)(1次元正規分布の検定)

②ホテリングのT^2検定(1標本問題)(Hotelling's one-sample T^2 test)(多変量正規分布の検定)

これらの統計的手段に関しては、例えば、参考文献[Takeuchi et al 1989]を参照されたし。

3 6 7 評価項目に対する重点テスト方法

評価項目は、3.1 1項に示されているように、下記に示すように分類される。

結果系:①プロダクトに関する評価(P)

要因系:②評価方法に関する評価(E)

③開発/保守方法・技法・ツールに関する評価(M)

その他:④経済性、社会性に関する評価(O)

②と③に対する評価は、主として(人手による)レヒューやインスペクションで行われる。④に関しては、本調査・研究では主たる対象としないのて、ここでも、適用対象としないことにする。

①に対する評価において使用されるテスト方法を表3 6-1に示す。表中の○印は、当該テスト方法を適用することにより、その評価項目の評価に必要とする直接的に使用可能なテスト結果を入手できることを示し、一方、△印は、当該テスト方法を適用することにより、その評価項目の評価に必要とする間接的に使用可能なテスト結果を入手てきることを示している。なお、○印や△印の付いていない評価項目に関しても、ここて示されているテスト方法を適宜組み合わせて適用することにより、その評価項目の評価に必要とする関連するテスト結果が入手可能となる。

表36-1 評価項目別の重点テスト方法（1／3）

評価項目 ＼ テスト方法	知識ベースの妥当性評価			問題解決・推論方式		結果の妥当性評価			要求仕様達成度評価		知識ベースの保守性評価				HIFの妥当性評価		システムの総合評価					その他		
	知識ベースの正当性評価	知識ベースの完備性評価	知識ベースの一貫性評価	推論処理の妥当性評価	推論時間の妥当性評価	正解の評価	解の網羅性評価	解の最適性評価	要求仕様内容の評価	要求仕様の達成度評価	変更箇所発見の容易性	知識ベースの理解性評価	変更箇所の局所性評価	知識ベースの統一性評価	利用時のI／F評価	保守時のI／F評価	テーマ選定の妥当性評価	利用者の満足度評価	統合性の評価	成長可能性の評価	開発保守用ドキュメント			
内容的妥当性テスト〈x-1〉	○	○	○	○		○	○	○	○								○							
基準に基づく妥当性テスト〈x-2〉	△	△	△	△	○	○	○	○		○								○						
構成概念的妥当性テスト〈x-3〉	○	○	○	○		○	○	○																
表面的妥当性テスト〈x-4〉	△	△	△	△	○	○	○	○							○	○		○						
未来予測的妥当性テスト〈x-5〉						○	○	○																
併存的妥当性テスト〈x-6〉						○	○	○																
過去予測的妥当性テスト〈x-7〉						○	○	○																
静的テスト／分析〈Y〉	○	○	○	○	△	△	△	△	○	○	○	○	○	○	○	○	○	○	○	○	○			
動的テスト〈Z〉	△	△	△	○	○	○	○	○		○					○			○						
レビュー〈Y-1-1〉	○	○	○	○	△	△	△	△	○	○	○	○	○	○	○	○	○	○	○	○	○			
ウォークスルー〈Y-1-2〉	○	○	○	○	△	△	△	△	△	○	○	○	○	○										
インスペクション〈Y-1-3〉	○	○	○	○	△	△	△	△	△	○	○	○	○	○										
変則評価〈Y-1-4〉	△	△	△	○							○	○	○	○										
リスク分析〈Y-1-5〉				△	△				△								△	△						
ハザード解析〈Y-1-6〉				△					△								△							
故障の木解析（FTA）〈Y-1-7〉				△					△								△							
ルールベースの論理的一貫性分析〈Y-2-1〉			○																					
ルールベースの論理的完備性分析〈Y-2-2〉		○																						
ルール間の相互参照分析〈Y-2-3〉	○	○	○	○																				
確信値の矛盾検出〈Y-2-4〉	○	△	△	○																				
データフロー分析〈Y-2-5〉	○	△	△	○																				
制御フロー分析〈Y-2-6〉	○	△	△	○																				

［注］HIF・ヒューマンインタフェースの略

表36-1 評価項目別の重点テスト方法（2／3）

評価項目 ＼ テスト方法	知識ベースの妥当性評価			問題解決・推論方式		結果の妥当性評価			要求仕様達成度評価		知識ベースの保守性評価				HIFの妥当性評価		システムの総合評価					その他		
	知識ベースの正当性評価	知識ベースの完備性評価	知識ベースの一貫性評価	推論処理の妥当性評価	推論時間の妥当性評価	正解の評価	解の網羅性評価	解の最適性評価	要求仕様内容の評価	要求仕様の達成度評価	変更箇所発見の容易性	知識ベースの理解性評価	変更箇所の局所性評価	知識ベースの統一性評価	利用時のI／F評価	保守時のI／F評価	テーマ選定の妥当性評価	利用者の満足度評価	統合性の評価	成長可能性の評価	開発保守用ドキュメント			
ランダムデータ法 <Z-1-1>						○	○	○										○						
同値分割法 <Z-1-2>						○	○	○																
限界値分析法 <Z-1-3>						○	○	○																
原因－結果グラフ法 <Z-1-4>						○	○	○																
層化抽出法 <Z-1-5>						○	○	○										○						
誤り推定法 <Z-1-6>						○	○	○																
運用プロフィール法 <Z-1-7>						○	○	○										○						
実データ法（本番データ法）<Z-1-8>						○	○	○										○						
標準テスト問題法 <Z-1-9>						○	○	○										○						
テスト網羅計測法 <Z-1-a>						△	△	△																
変異テスト法 <Z-1-b>						△	△	△																
ﾎﾜｲﾄﾎﾞｯｸｽﾃｽﾄ/構造ﾃｽﾄ<Z-2-1>	○	○	○	○		○	○	○		△	△	△	△	△										
ﾌﾞﾗｯｸﾎﾞｯｸｽﾃｽﾄ/機能ﾃｽﾄ<Z-2-2>					○	○	○	○		○														
回帰テスト <Z-3-1>				△	△	△	△	△												△				
フィールドテスト <Z-4-1>					○	○	○	○		○								○	△	△				
実験室テスト <Z-4-2>				○	○	○	○	○																
負荷テスト <Z-5-1>					○					○								○						
感度分析テスト <Z-5-2>	○	○	○	○	○	○	○	○																
削除テスト <Z-5-3>		○								○														
説明機能 <Z-6-1>	△	△	△	○		△	△	△																
プロトコル分析（内観分析）<Z-6-2>	△	△	△	○	○	△	△	△							○	○		○						

［注］HIF　ヒューマンインタフェースの略

表3 6-1 評価項目別の重点テスト方法（3／3）

評価項目 ＼ テスト方法	知識ベースの妥当性評価			問題解決・推論方式		結果の妥当性評価			要求仕様達成度評価		知識ベースの保守性評価				HIFの妥当性評価		システムの総合評価					その他		
	知識ベースの正当性評価	知識ベースの完備性評価	知識ベースの一貫性評価	推論処理の妥当性評価	推論時間の妥当性評価	正解の評価	解の網羅性評価	解の最適性評価	要求仕様内容の評価	要求仕様の達成度評価	変更箇所発見の容易性	知識ベースの理解性評価	変更箇所の局所性評価	知識ベースの統一性評価	利用時のI／F評価	保守時のI／F評価	テーマ選定の妥当性評価	利用者の満足度評価	統合性の評価	成長可能性の評価	開発保守用ドキュメント			
過去／既存のデータ／結果<Z-7-1>						○	○	○		○								○						
1人の専門家<Z-7-2>	△	△	△	△		○	○	○		○					△		△	○						
専門家のパネル（複数の専門家）<Z-7-3>	○	○	○	○		○	○	○		○					△		△	○						
対照基準グループ<Z-7-4>										○								△						
併行運用<Z-7-5>						○	○	○		○							△	△						
ノミュレーノョノ<Z-7-6>						○	○	○																
他のモデル<Z-7-7>						○	○	○									△	△						
実　験<Z-7-8>						○	○	○																
実地調査<Z-7-9>						○	○	○																
専門家間の一致係数<Z-8-1>						△	△	△																
信頼区間<Z-8-2>						△	△	△																
第Ⅰ種の誤り<Z-8-3>						△	△	△																
第Ⅱ種の誤り<Z-8-4>						△	△	△																
ﾁｭｰﾘﾝｸﾞ ﾃｽﾄ/目かくし評価<Z-9-1>						△	△	△																
クロスチェック<Z-9-2>						△	△	△																
統計的手段<Z-9-3>						△	△	△																

［注］HIF　ヒューマンインタフェースの略

《参考文献》

[ANSI/IEEE 1983] 東基衛(監修), ＡＮＳＩ／ＩＥＥＥソフトウェア規格集, 日本規格協会, 1988年2月.

[Bell 1989] Bell, T.E., "Managing Murphy's Law: Engineering a Minimum-Risk System," IEEE Spectrum, Vol.26, No.6, pp.24-27, (June 1989).

[Benbasat & Dhaliwal 1989] Bendasat, I. and Dhaliwal, J.S., "A Framework for the Validation of Knowledge Acquisition," Knowledge Acquisition, Vol.1, No.2, pp.215-233, (June 1989).

[Deutsch & Willis 1988] Deutsch, M.S. and Willis, R.P., (成田訳), ソフトウエア品質工学, 日経ＢＰ社, 1990年5月, pp.293.

[Gaschnig et al. 1983] Gaschnig, J.G., Klahr, P., Pople, H., Shortliffe, E., and Terry, A., "Evaluation of Expert Systems: Issues and Case Studies," in Building Expert Systems, Hayes-Roth, F., Waterman, D.A., and Lenat, D.B., Eds, Addison-Wesley, Reading, pp.241-280, 1983.

[Green & Keyes 1987] Green, C.J.R. and Keyes, M.M., "Verification and Validation of Expert Systems," WESTEX-87, pp.38-43, (June 1987).

[Ikeda 1973] 池田央, 心理学研究法８ テストⅡ, 東京大学出版会, 1973年7月, pp.324.

[Ishii 1986] 石井康雄(編), ソフトウェアの検査と品質保証, 日科技連, 1986年10月, pp.240.

[Leveson 1986] Leveson, N.G., "Software Safety: Why, What, and How," ACM Computing Surveys, Vol.18, No.2, pp.125-163, (June 1986).

[Levinson & Daughtrey 1990] Levinson, S.H. and Daughtrey, H.T., "Issues in Verifying Nuclear Power Plant Expert Systems," Proc. of the Topical Meeting on Advances in Human Factors Research on Man/Computer Interactions: Nuclear and Beyond, pp.33-39, (June 1990).

[Miller 1989] Miller, L.A., "Validation and Verification of Knowledge-Based Systems," Invited Tutorial at the Conference on Expert Systems Applications for Electric Power Industry, June 5, 1989, pp.22.

[Miura et al. 1985] 三浦, 狩野, 津田, 大橋(編), ＴＱＣ用語辞典, 日本規格協会, 1985年1月, pp.571.

[Musa 1989] Musa, J.D., "Tools for Measuring Software Reliability," IEEE Spectrum, Vol.26, No.2, pp.39-42, (Feb. 1989).

[Nguyen et al. 1987] Nguyen, T.A., Perkins, W.A., Laffey, T.J., and Pecora, D., "Knowledge Base Verification," AI Magazine, Vol.8, No.2, pp.69-75, (Summer 1987).

[O'Keefe et al. 1987] O'Keefe, R.M., Balci, O., and Smith, E.P., "Validating Expert System Performance," IEEE EXPERT, Vol.2, No.4, pp.81-90, (Winter 1987).

[O'Keefe & O'Leary 1989] O'Keefe, R.M. and O'Leary, D.E., The Verification and Validation of Expert Systems, IJCAI-89 Tutorial Program MP4, Aug. 21, 1989, pp.31.

[O'Leary 1987] O'Leary, D.E., "Validation of Expert Systems - with Applications to Auditing and Accounting Expert Systems," Decision Sciences, Vol.18, No.3, pp.468-486, (Summer 1987).

[O'Leary,T. et al. 1990] O'Leary, T.J., Goul, M., Moffitt, K.E., and Radwan, A.E., "Validating Expert Systems," IEEE EXPERT, Vol.5, No.3, pp.51-58, (June 1990).

[Pressman 1982] プレスマン, D.S., (岡田訳), ソフトウェア・エンジニアリング序説, ＴＢＳ出版, 1983年7月, pp.361.

[Rossomando 1989] Rossomando, P.J., "A Knowledge Based Environment for the Verification and Validation of Knowledge Based Systems," Preliminary Proceedings of the Workshop on Verification, Validation and Testing of Knowledge-Based Systems at IJCAI-89, pp.15, (Aug. 1989).

[Takeuchi et al. 1989] 竹内啓他(編), 統計学辞典, 東洋経済新報社, 1989年12月, pp.820-823.

[Tamai et al. 1988] 玉井, 三嶋, 松田, ソフトウェアのテスト技法, 共立出版, 1988年1月, pp.182.

[Tsai & Zualkerman 1990] Tsai, W.T. and Zualkerman, I.A., "Towards a Unified Framework for Testing Expert Systems," Proc. of the Second International Conference on Software Engineering and Knowledge Engineering, pp.127-134, (June 1990).

[Walters & Nielsen 1988] Walters, J. and Nielsen, N.R., Crafting Knowledge-Based Systems: Expert Systems Made Easy Realistic, John Wiley & Sons, Reading, New York, pp.342, 1988.

[Wright 1990] Wright, R.G., Expert System Verification and Validation, IEEE Managing Expert System Programs and Projects Conference Tutorial#2, Sept. 10, 1990, pp.116.

[Yamaguchi et al. 1987] 山口, 溝口, 田岡, 小高, 野村, 角所, 「深い知識に基づく一般ユーザー向けの説明機能」, 電子情報通信学会論文誌Ｄ, Vol.J70-D, No.11, pp.2083-2088, (1987年11月).

4　本ガイトラインの適用と評価

4.　本ガイトラインの適用と評価

本章では、第３章で提案したエキスパートシステム評価ガイトラインの有用性を評価するために、本研究委員会委員が実際に携わったエキスパートンステムを事例として、本カイトラインを適用することとした。

ここで対象としたエキスパートシステムは、合成型問題として、ジョブンョップスケジューリング問題とＬＳＩ論理合成問題の２つ、解析型問題として、鉄鋼プラント異常診断問題と水力鋼構造物評価診断問題の２つ計４つを取り上げる。これらのシステムはいずれも実運用型エキスパートシステムとして開発されたものである。しかし、これらは、既に完成したシステムであるので、ガイドラインを開発中に適用することはできなかった。そこて、カイトラインの分析に当たっては、開発途中のエキスパートンステムに適用することを想定して、どのような効果・欠点が考えられるかを十分に考慮することとした。

また、ガイトライン自体を適用した結果、ガイドライン自体が有用てあった点と今後の課題についても取りまとめる。

本章の目次構成は、以下の通りである。

4.1 ジョブショップスケジューリングシステムへの適用
4.2 鉄鋼プラント異常診断エキスパートシステムへの適用
4 3 ＬＳＩ論理合成エキスパートシステムへの適用
4.4 水力鋼構造物評価診断システムへの適用
4 5 ガイトラインの評価

4.1 ジョブショップスケジューリングシステムへの適用

4.1.1 プロフィール

4.1.1.1 システムの目的と期待効果

(1) 問題の背景

ジョブショップスケジューリングの分野では、従来からＰＥＲＴ（Program Evaluation and Review Technique）等のＯＲ手法による計画が行われてきた。しかし、これらの手法の問題点として解を出すためのモデルが非常に簡単化されていて、実際に計画担当者が行っている様な複雑な制約や政策的意図を反映した計画を作成することは出来なかった。また、計画の規模もダイナミックさも要求されるようになり、エキスパートシステムでの対応に期待がかけられてている。

(2) システムの目的とその期待効果（要求仕様）

システムの目的とその期待効果（要求仕様）は、以下の通りである。

- 省力化（専門担当者：３人×２日→専門担当者の廃止、0.5人日）
- ダイナミックな計画変更（0.5～２時間／件）
- 知識の共有化
- 環境変化への柔軟な対応
- 計画の質的向上（中間在庫の削減、マシンバランスの適正化）

4.1.1.2 システムの概要

(1) システム対象

本システムは多品種少量生産の部品加工ラインのスケジューリングであり、日程計画を作成する。

(2) ジョブショップとは

ジョブショップとは、注文品（ジョブ）が複数の工程（ショップ）を経由して、完成品になる形態のことをいう。ジョブはどの様な工程を経て行くべきかはジョブ毎に規定されている。このことは、全てのジョブが決まった工程を経由していくフロォーショップと対応される。逆に言うとフロォーショップはジョブショップの特殊型ということが出来る。従って、ジョブショップのスケジューリングにおいては、いかにショップ内での段取り工数を極力少なくし、ショップ間の渡りのための待ち（仕掛かり在庫）を少なくし、ショップ間の負荷バランス、ショップ内での負荷バランスを取りながら納期をキープするスケジュールを立てることがポイントである。

(3) システムの規模

システムの規模は、以下の通りである。

- 生産品種：300品目
- 平均ロットサイズ：50個
- ジョブ数：200件／月
- 平均工程数：４５工程／ジョブ
- ショップ数：８ショップ

• 1ショップ当りの平均マシン数：3.5台／ショップ

(4) ジョブショップスケジューリングＥＳの流れ
本システムの処理の流れの概略は、図4.1-1の通りである。

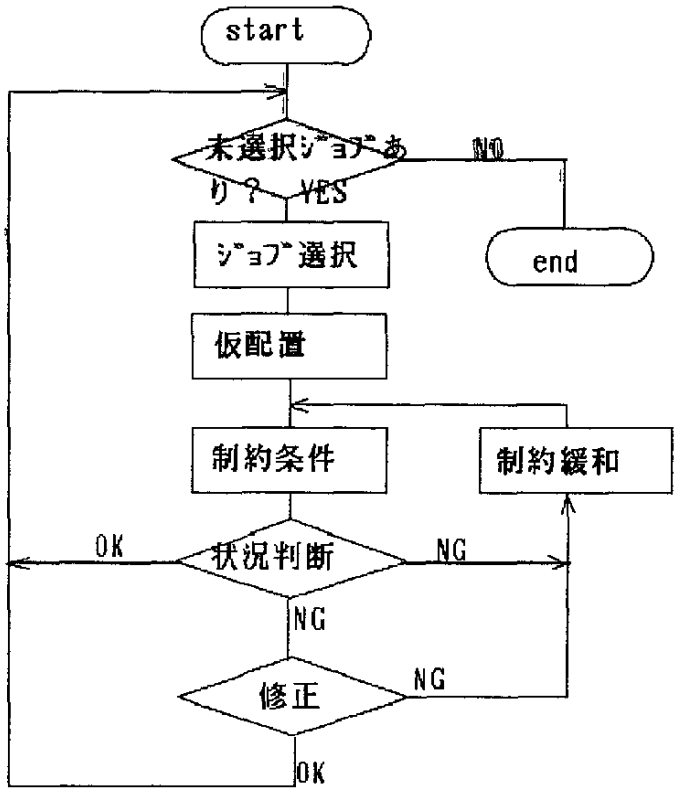


図4.1-1　処理の流れ

4.1.1 3　開発フェーズと評価体制

(1) 開発フェーズ
本システムの開発ライフサイクルでの位置づけは、実用システム開発フェーズに相当する。

(2) 評価体制
本システムの評価は以下のような役割分担て行った。
• 専門家（＝運用担当者）
　• テストデータの用意
　• テスト結果の解析・評価
　• ヒューマンインタフェースの評価
• レビューア（＝開発担当者）
　• 推論効率の評価

- 専門家のテスト結果の解析を受けて推論トレースの分析
- 知識の冗長性等の評価

- マネージャ
  - 要求仕様の達成度評価
  - システムの総合評価

4 1 2 ガイトラインの適用結果

本システムをガイトラインに沿って評価していくこととする。特に計画型問題において注目すべき評価項目に対して重点的に評価をする。

(1) 知識ベースの妥当性評価

(1-1) 知識ベースの正当性評価

(C01)知識源の確認

本件のケースでは、３人の生産管理担当者がいるが、本当の専門家は１人で辣腕をふるい他の担当者は、それを補佐するという役割てあり、知識はその専門家から収集した。従って第３者専門家というものを探すのは困難であった。制約の知識は、かなり論理立ったものが多く、その理由を求めることによって、その正当性はある程度確証できた。制約緩和知識は、かなり政策的（例えば休日出勤、マシーンの定期検査等）なことがあり、その正当性を求めることは困難であった。

(C03)知識の整理

知識分類を以下のようにした。

- 制約知識：JOB間、JOB－ﾘｿｰｽ間、JOB－時間、ﾘｿｰｽ－ﾘｿｰｽ間、ﾘｿｰｽ－時間
- 制約緩和知識
- 評価知識
- 推論制御知識

(1-2) 知識ベースの完備性評価

(C05)レベルのアンバランス

計画型問題の場合、制約知識などは、ルール連鎖を静的に書くことは難しい。従って、完備性の評価は動的テストに多くは頼らざるを得ない。感度分析(C06)により、利用価値の少ないルールは見い出せても、足りない知識を出すことはできない。従ってシステムが出力した解を分析し、その配置がおかしいと指摘された所を中心に制約条件の不備を抽出する方法をとった。テスト結果は、以下の通りである。

＜20事例で最初の回答に対する専門家の評価＞

- 非常に満足：10%
- まあまあ　：60%
- やや不満　：15%
- 不満　　：15%

(1-3) 知識ベースの一貫性評価

(C01)外部環境の変化への対応

外部環境要素を知識として外に出して、変更可能なようにした。

(C05)モジュール化

知識は、(1-1)のように分類・体系化し、見通しを良くした。

(C08)ルールの条件部の適切さ

制約知識には「禁止、原則ルール」「推奨、制止ルール」を設け、その制約の厳しさをルール優先度でコントロールしている。この優先度の妥当性の検証は難しい。

(C11)フレームの一貫性のテスト

チェックは、ツール側て行われている。

(1-4) 経験的知識を正当化する方法とその確認

(C05)過去のデータからの検討

(1-2) の通り過去の事例を用いてテストして正当性の確認を行った。

(1-5) 知識表現の選択と適切さの確認

(C01)知識表現方法と対象問題の特徴の関連

制約条件はフレーム形式で表現した。これは、制約のクラス化が行い易く、また可能性に富み、メインテナンスが行い易いという判断から、フレームを採用した。

推論制御ルールは、プロダクション及びメソッドで表現した。これは、このメインテナンスはユーザではなく開発部門のＫＥが行うという前提で高速性を重点において決めた（推論制御ルールで用いる外部環境は、フレーム、テーブル形式で外に出した）。

(1-6) 知識獲得方法の選択と適切さの確認

(C01)専門家の選定方法の妥当性評価、(C03)知識の収集

知識獲得は、一般のインタビューによる獲得と実際に専門家がスケジュールを立案している時に立ち合いインタビューする方法の２つを採用した。一般のインタビューでは、表層的な知識しか収集されなかったが、立ち合い時のインタビューの時は、その妥当性を示すような深い知識の獲得もできた。

(2) 問題解決・推論方式の妥当性評価

(2-1) 推論処理の妥当性評価

(C02)モジュール化

スケジュール問題の基本タスクを分析し、その単位で、推論機構を構築しているため、各基本タスク毎のアウトプットを分析することにより、推論の妥当性がチェックしやすくなっている。

(C08)推論の停止条件

外部環境の変化に関係するものは、推論制御ルールから外出しにした。

(2-2) 推論時間の妥当性

(C01)専門家のイメージ

専門家の処理時間の約1/10倍かかる。

(M04)生成検査法

一つのノードの生成にかかる平均処理時間は、以下の通りてある。

投入ジョブ200件、平均工程数4 5工程／ジョブとする。

- ノード数＝200×4 5＝900ノート
- 1つの計画案を出す時間：平均約30分　　注）計画が決定するまでの時間ではない
- ノート当りの平均処理時間＝900÷1800＝0 5秒／ノート

(2-3) 推論手法の選択と適切さの確認

(C02)基本的な推論手法

生成検査法を採用している。

(2-4) 不確実情報に対する処理技法の適切さの確認

(C05)確信度の性質

ルールには「～しないほうが良い」、「～したほうが良い」というような曖昧な表現がある。これらのルールに重み付けを行って対処した。しかしこの重み付けにはっきりした根拠は存在しなかった。

(2-5) 学習機能の適切さの確認

(3) 結果の妥当性評価

(3-1) 正解の評価

(C01)正解の定義

スケジューリング問題の場合は、解が正解、不正解と明確に区別されるのではなく、より良い解という見方がなされる。従って厳密な正解を求める必要はない。解の良し悪しは、専門家の模範解と比較して、その性能を評価した。しかし、逆に専門家の立案は必ずしも全ての制約を満足しているとは限らないので、計算機からの解の方が良い場合もあった。

(3-2) 解の網羅性評価

(C01)網羅性の目標

最適解てはなく、よりよい解を求めるンステムてある。従って、1つずつ出てきた解をユーザに確認してもらう方式をとっている。推論時間が設定できるようになっており、その時間に解が出ない時は、途中結果を出力し、後は利用者にまかせる機構を採用している。

(3-3) 解の最適性評価

(C02)評価関数がある場合

ラインバランスと納期遅延率を評価関数にとっている。

(C08)コストとのバランス

解の最適性は追求しないて、時間内に求まるより良い解を追求する。

(3-4) 解の評価方法の適切さの確認

(C01)評価方法

知識提供の専門家による評価（専門家の能力と比較）と、レヒューアによる解の確認を選択した。

(4) 要求仕様に対する達成度評価

(4-1) 要求仕様内容に対する評価

(C06)実現可能性

要求仕様の主な内容は、処理時間、解の満足度（専門家が評価し平均満足度60%以上）、省力効果である。

(4-2) 要求仕様に対する達成度評価

(*C1)ユーザの要求の達成度

下記のユーザの要求は、システムによりほぼ達成できた。

- システム化の目的
- プロジェクト要件（開発費、納期、要員）
- 機能要件（要求される機能、想定したデータの範囲、出力結果など）
- ユーザ要件（ユーザインタフェース、導入効果）
- 性能要件（応答時間、正確性、保守性、使用容易性）
- 環境要件（資源、ハートウェアの制約）

(4-3) 要求仕様設定手段の確認

(5) 知識ベースの保守性評価

(5-1) 変更箇所発見の容易性評価

(C01)(C02)(C03)モジュール化

知識のブロック化、基本タスクの抽出とモジュラリティを計っている。

(C04)トレース機能

ルール連鎖のトレースはとれるが、スケジューリング問題ではあまり有効ではない（そのルールが発火したスケジュールの状態と対になっていないと有効ではなく、その状態を含めてもつことは、メモリの制約上、現実的ではない）。

(C07)名称の分かりやすさ

分かり易いルール名をつけることはなかなか難しかった。

(5-2) 知識ベースの理解性評価

(C01)専門家による保守

保守の可能性の高い知識はフレーム化した。

(C02)モデルの素直さ

専門家が初期から参加した。

(5-3) 変更箇所の局所性評価

(C02)変更の少ない知識と多い知識

知識を分析し体系化した。その結果、変更されやすい知識とそうでない知識の区別が明確になった。

(M01)問題あたりの修正ルール数

特に測定はしていない。

(5-4) 知識ベースの統一性評価

(5-5) 品質安定性の確認方法の確認

(C02)動作確認用のサンプルデータの有無

特にサンプルデータは用意していない（スケジューリング問題の場合、単一解ではないのでサンプルデータの考え方がなかなかそくわない）

(C03)バージョン管理

知識にはメインテナンスの日付・担当名が記入できるようになっている。

(5-6) 保守マニュアルの整備状況の確認

(C01)保守マニュアル、(C02)文書化基準

分かりやすいマニュアルになっているかは疑問である。

(C06)保守履歴

上記(5-5)の通りである。

(5-7) 保守用ツールの整備状況の確認

(C01)ツールの整備

知識エディタ、説明ツールは用意しているが、解析ツール、テストツールは用意されていない。用意されているツールも充分であるとは言えない。

(6) ヒューマンインタフェースの妥当性評価

(6-1) 利用時のインタフェース評価

(C01)結果の表示

計画表へ配置状況を表示する（これが利用者にとって役に立っているかは疑問）。

(C02)入力の操作性

ユーザが推論結果をマウスを使って簡単に修正できる機構を提供している。

(C04)ＨＥＬＰ機能

ＨＥＬＰガイダンスが充分であるとは言えない。

(C06)習熟度

利用者の習熟度に応したガイダインスにはなっていない。

(C13)ＷＨＹ機能

スケジューリングの場合、そこに配置した“ｗｈｙ”を説明する理由は、１つは「なにも制約がなかったから置いた」（積極的理由）という理由と、「本当は違う所に置きたかったが、しかたなく置いた」（消極的理由）の２つがある。後者を説明することは難しい。

(6-2) 保守時のインタフェース評価

(7) システムの総合評価

(7-1) テーマ選定の妥当性評価
(C01)ニーズの強さ
作業効率の向上、専門家の作業支援である。
(C02)明示的な知識
今までも従来の手法で何回も試みたが、うまくいかなかった。

(7-2) 利用者の満足度評価
(C03)導入効果、利点
従来の1/10に作業が短縮した。またこのシステムの導入により月中でも、ダイナミックな計画変更が行えるようになった。

(7-3) 統合性の評価
(C02)既存機能または既存システムとの連続性
オーダエントリシステムと結合し、オーダ情報が入力としてとれるようになっている。将来は、各ショップの部品在庫情報から発注システムへのリンクを考えている。

(7-4) 成長可能性の評価
(C01)レビュー
何が外部環境の変化により変わる要素かを分析し、そのような項目はアルゴリズムに組み込まず、外に出して変更できるようにした。また、その限界もオーソライズした。

(7-5) 開発／保守用ドキュメントの評価
(C01)～(C08)ドキュメント全体
必ずしも満足のいく状態ではない。
(C09)～(C15)ワークシート関連
ワークシートは記憶を呼び戻すトリガーにはなっても、全てそれによってたどれるかというとそこまで完全には記述、及び保管がされていない。

(7-6) 開発方法の適切さの確認
(C01)体系的な開発方法
調査・計画、プロトタイプ、実用化とライフサイクルに沿って開発した。

(7-7) ツール・言語選択の適切さの確認
(C03)高性能のツール選択
自前のツールがあるため、ツールありきから始まっている。しかし、結果的には、それほど無理はなかった。

(7-8) システム開発の生産性評価
(C01)生産性指標
時間的には見積りとおりてあったが、コスト（工数）は見積の約10％ｕｐであった。

(7-9) システムのコストパフォーマンス評価

(C01)経費の節減

直接的には効率が10倍良くなった。間接的には月中てのダイナミックな計画の修正が行いやすくなったこと、及び計画担当者の専任を置かなくても計画立案ができるようになった。

(C07)運用コスト

運用コストは未だ評価していない(これから先、知識メインテナンスかとのくらい発生するかは不明てある)。

(7-9) セキュリティ対策の評価

(C01)〜(C09)特に講じていない。

4.2　鉄鋼プラント異常診断エキスパートシステムへの適用

4.2 1 プロフィール

4.2.1 1 システムの目的

溶鋼を凝固させて鋳片を製造するための鉄鋼プラント（連続鋳造設備）においては、たまに、溶鋼が完全に凝固しない状態で鋳片が引き出され、溶鋼が鋳片から噴き出すという異常状態「ブレイクアウト」が発生し、プラント操業に大きな影響を与えることがある。本エキスパートシステムでは、このブレイクアウトを検知し、未然にブレイクアウトを防ぐことが目的である。

4 2.1 2 問題の特定（タスク構造の決定）

本プロジェクトは、以下に示す５名の構成員によって進められ、タスク構造を決定するために、「入力」、「処理」、「出力」について議論が重ねられた。

ユーザサイト　　：　システムマネージャ１名、専門家２名
システム開発サイド　：　プロジェクトリーダ兼ＫＥ１名、
　　ＫＥ兼プログラマ１名

入力については、鉄鋼プラントの状態を把握するためのファクタとして、①鋳型内の温度、鋳片を引き抜く鋳造速度、鋳型内の溶鋼の湯面レベル（位置）を測定するためのセンサー、②溶鋼の成分、③操業条件などが議論されたが、②と③は明確なデータとして入力することが困難であるため、当面、①のみを入力データとすることにした。

処理としては、温度センサーからのサンプリングデータの解釈と、温度センサーと他のセンサーとの関連による操業条件の把握を通して、ブレイクアウトの前兆を認識することにした。但し、ブレイクアウトが現実に起こる前に、システムが判断を下す必要があるため、１０数秒程度で認識しなければならない。

出力については、本エキスパートシステムが鋳造速度の制御機器を直接制御することが最終目標ではあるが、運用を通してシステムの信頼性を確認した後に、その実行可能性を判断することとし、当面は、プラント操作員を支援する形て運用することにした。

4 2 1 3 システムの概要

(1) システムの概観

本エキスパートシステムの構成を図4.2-1 に示す。本エキスパートシステムは、通常の前向き推論を行うエキスパートシステム（実行部）と、ルールを逐次的に洗練化していく知識洗練化部に大きく分かれる。以下に、重要なシステム構成要素の概略について述べる。

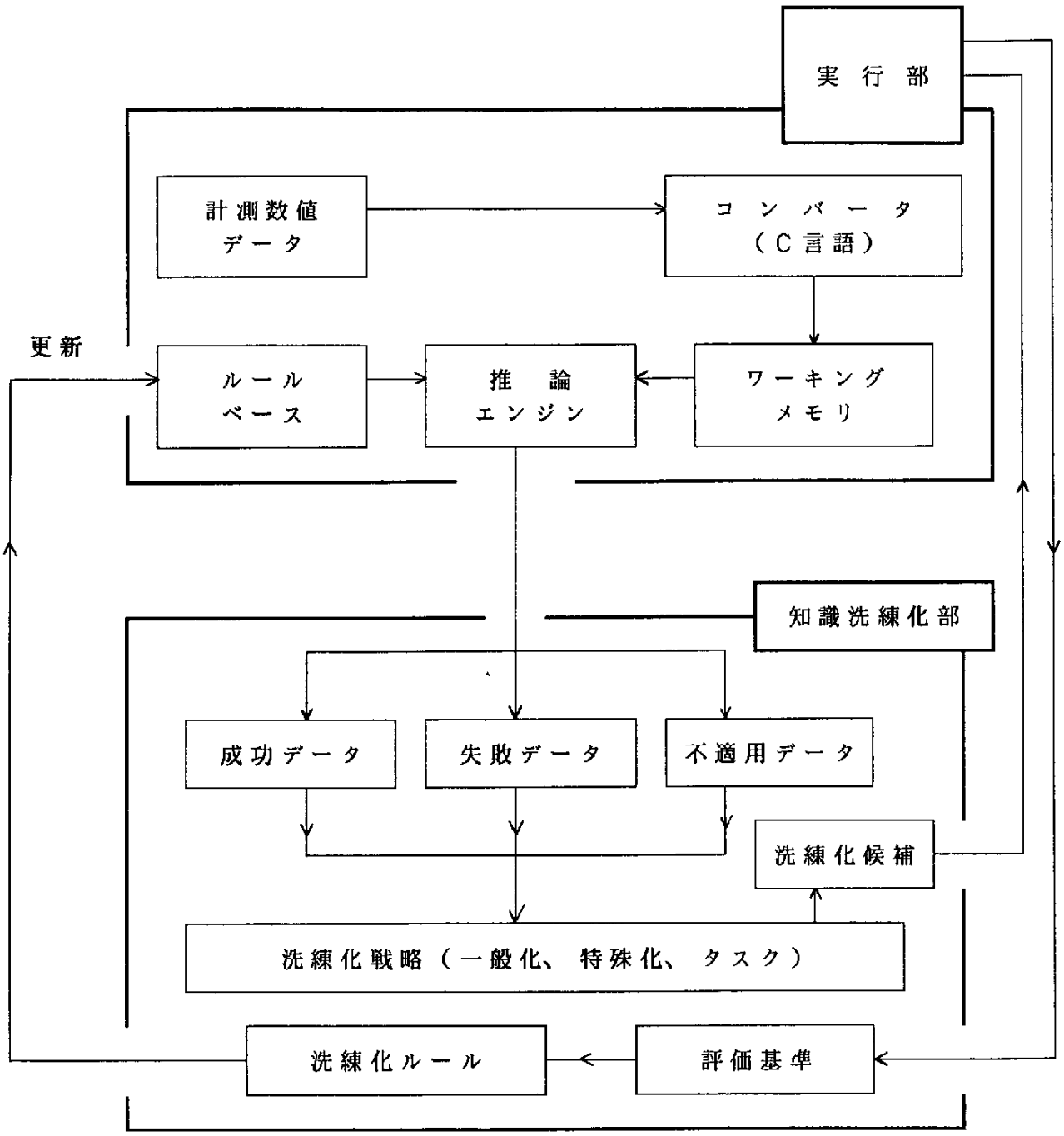


図 4 2-1 システムの構成

(2) コンバータ

コンバータは、センサー（鋳造速度センサー＋湯面レベルセンサー＋８個の温度センサー）から逐次与えられる１０秒間の時系列数値データについて、変化の傾向（トレンド）、１トレントの持続時間、１トレントの振幅幅を記号データに変換して、ワーキングメモリに蓄える。

トレンドは、７段階（激しく増加：i3 ～ 定常：c ～ 激しく減少 d3 ）に分けて記号化されるが、トレンドをマクロに捉えることが重要である。代表的なト

レントのセグメンテーション法を以下に示す。

①持続時間が0 1秒のトレントは除去する。ただしその微小変化が0 6秒以上続いた場合はそれらの傾きの平均をとって1つの傾きとする。従って、0 1秒の微小変化の連続が5サイクル以内であればそれらの傾きは除去する。

②あるトレントの持続時間が0 2秒で、その両側のトレントが同種のトレントで持続時間が0 4秒以上である時、1つのトレントにまとめる。

この他、分析時点より過去一定時間内でのd3の出現回数と、同一分析時点において同じ特徴変化を示す異なる温度センサーの数も調べる。

(3) ルールベースと推論エンジン

ブレイクアウト診断の経験的知識がルール化されているが、知識洗練化部により、ルールは逐次的に洗練化されていくため、それらの記録情報も蓄える構造になっている。推論エンジンは、前向き推論を行うための簡単なものであり、ワーキングメモリとルールの条件部とのマッチングの成功時に、ブレイクアウトが診断され、マッチングの失敗は、鉄鋼プラントが正常であるという判断になる。

(4) 知識洗練化部

(4-1)基本的考察

ルールの質を向上させるための知識洗練化機構については、初期のルールを適用して得られた実験データを分析することにより考察した。

即ち、実験データを表4 2-1 のように分類し、それらのデータ間の差異を比較検討した。

表4 2-1 実験データの分類

現実	ｴｷｽﾊﾟｰﾄｼｽﾃﾑ	データの分類
正常	正常	成功データ
正常	ﾌﾞﾚｲｸｱｳﾄ	失敗データ
ﾌﾞﾚｲｸｱｳﾄ	正常	不適用データ
ﾌﾞﾚｲｸｱｳﾄ	ﾌﾞﾚｲｸｱｳﾄ	成功データ

その結果、以下に示すようなデータ(temp)がワーキングメモリ中に多数存在することが判明した。

成功データ： temp(4,38,i3,29,15 4,67,i1,6,0.2,73,d3,25,16 6)
失敗データ： temp(3,42,i3,34,13 6,68,c,2,0,72,d3,28,10 8)

これらのデータに対しては、下線部で示したトレントi3の持続時間を一般化（長く）あるいは特殊化（短く）することにより、対処できる。

また、以下に示すようなデータもワーキングメモリ中に存在した。

成功データ ： temp(4,38,i3,29,15 4,67,i1,6,0 2,73,d3,25,16 6)
失敗データ ： temp(3,17,i3,26,10 4,56,c,4,0,64,d3,31,12 8)
不適用データ： temp(3,17,i3,14,10 4,56,c,4,0,64,d3,31,12 8)

上記のデータの場合、初期のルール構造では、不適用データとはマッチして、

失敗データとはマッチしないので洗練化はできない。

しかしながら、成功データと不適用データには、この分析時点よりも前の変化に注目すると、非常に不安定な温度変化が見られるのに対して、失敗データの方はさほど不安定な温度変化はみられないことが、実際のデータをよく観察してみると判明した。そして、この種の新しいプリミティブは、専門家からは教示されなかったけれども、パターン認識で比較的よく使われる特徴量に関連するものか多く、その利用を試みることにした。

(4-2)システム構成

知識洗練化部では、まず、実行部で利用されたデータを、成功データ、失敗データと不適用データに分類する。

次に、一般化戦略ではルールと不適用データを、特殊化戦略ではルールと成功データと失敗データを、タスク戦略ではルールと成功データと失敗データをそれぞれ参照比較し、洗練化候補ルールを生成する。

さらに、生成された洗練化候補ルールを実行部で実行し、最適な洗練化ルール候補を決定すると共に、知識ベース全体で洗練化前よりも性能が向上していれば、この洗練化ルール候補を新しいルールとしてルールベースに蓄える。

(5) 一般化・特殊化戦略

基本的に、一般化・特殊化戦略としては、「ルール中のプリミティブの値を変数化・定数化する戦略」を採用している。

一般化戦略では、トレントの持続時間の制限範囲を拡張することにより、ブレイクアウトとなるへき不適用データを成功データに加えることを試みる。但し、ブレイクアウトを示すデータがすへて認識されない時に適用される。

特殊化戦略では、成功データと失敗データを比較し、トレントの持続時間の制限範囲を短縮することにより、失敗データを減らすことを試みる。但し、特殊化戦略を適用することによって成功データが減少するケースもあり、ブレイクアウト発生時のデータの網羅率を下げない場合に限り、適用される。

(6) タスク戦略

トレント分析は、波形パターン認識と密接な関連があるはすてあり、たとえ専門家が言及していなくても、波形パターン認識で使用される汎用な特徴量（属性）を用いて、ルールが洗練化できないかどうかを試みる。

本戦略では、その特徴量として、①振幅、②分析時点より過去一定時間内てのある特徴変化の出現回数、の２種類を現在考えている。そしてこれらの特徴量を逐次ルールに付加して洗練化を試みる。即ち、これらの新しい特徴量（プリミティブ）を導入すれば、それまて表現されていなかった成功データと失敗テータ間の差異が記号レヘルに現れる可能性があるのてある。

(7) 動作環境

プロトタイプンステムは、SONY NWS-1850上で、コンバータはＣ言語、それ以外のモジュールはProlog言語で構築した。

知識洗練化部は、推論時間にそれ程制約がないが、実用ンステムの実行部に対しては、約１０数秒以内にブレイクアウトを診断しないと役に立たないという、準リアルタイム性の制約が課せられるため、Prologを主実装言語とした実行部て

は対処しきれなかった。そこで、実用システムの実行部は、すべてC言語により書き換えた。

4 2 1 4 問題のタイプとシステムの特性

本エキスパートシステムは、診断と解釈の両特性を保持しているが、タスクとしては異常診断に重点が置かれているので、診断型システムの特性と本エキスパートシステムの特性との関連性・実現性を表4.2-2 に示す。

表4 2-2 システムの特性一覧表

	診断型システム	関連性	実現性	コメント
定義	対象とするシステムに異常または故障が生したとき、観測されたデータ及びシステムに関する知識を利用して、原因箇所を同定することである。			
問題の特徴	1 設計レベルの知識の利用	○	△	知識の改訂時に利用した
	2 経験的知識の利用	○	○	
	3 経験的知識の不確実性	○	○	不確実性は、知識洗練化機構により対処した
	4 操業データの利用	○	○	
	5 計測のための時間とコスト	○	○	
	6 知識の抽象化の必要性	×	×	
	7 対話的診断の推論制御	×	×	
基本タスク	1 異常現象の分類階層的表現	△	×	
	2 システム構造の階層的表現	○	×	
	3 解釈問題のタスクを内包	○	○	
	4 計測点の選択的決定	×	×	
	5 異常原因の同定	○	×	
	6 浅いモデルによる効率性	○	○	計測時から10数秒以内でアクション をとる必要あり
	7 深いモデルによる完全性	○	△	
問題解決機能	1 事象駆動型推論	○	○	
	2 目標駆動型推論	×	×	
	3 不確実性推論	○	△	
	4 仮説推論	○	×	
	5 協調型推論	×	×	
	6 効率性と完全性の調和	△	×	
	7 費用／効果分析	○	△	

4 2 2 ガイトラインの適用結果

3 1 節で述へた点検項目（C(-) の記号）と測定項目（M(-) の記号）の中で、関連する項目から本システムを評価した。なお、(C01) なとの番号は3 1 節て述へたものに対応する。

(1) 知識ベースの妥当性評価

(1-1) 知識ベースの正当性の評価

(C01)経験的知識の根拠

本エキスパートシステムのルールは、「IF 計測データ　THEN ﾌﾞﾚｲｸｱｳﾄ」という形式であるが、ブレイクアウトはそれ程頻繁に起こる事象てはないため、ルール（条件部と結論部の関連性）の不確実性が問題になった。そこで、ルールの根拠として、今までに整理されたブレイクアウト発生メカニスムを調査し、その関連からルールを再検討した。

(C04)専門家によるレビュー、(C07)専門家間での合意

経験的知識は、２人の専門家からインタビューして獲得し、獲得された知識のレビューは、過去の計測データによりその知識を作動させた上で行った。

また、知識の解釈が異なる場合は、できるだけ、上記の発生メカニズムレヘルに立ち戻って、合意を得るように努めた。

(C10)結果の視覚化

計測データの変化状況をグラフ化するインタフェースを備えることにより、システムの振舞いを専門家が確認しやすくした。

(1-2) 知識ベースの完備性評価

(C07)完備性のテスト

過去の測定データを何回も取り替えて実験することにより、ルールの不備（特に、重要な属性の欠如）が発見できた。また、属性値の設定については、知識洗練化機構に任せる枠組みを採用した。

(1-3) 知識ベースの一貫性評価

(C08)ルールの条件部の適切さ

適切なルールの条件部の設定は、ブレイクアウトがそれ程頻繁に起こる事象ではないことから困難であり、実験結果を検討して、その度に人間が介在して、逐次ルールを修正する環境では非効率的てあることから、前述の知識洗練化機構により、対処することにした。

(1-4) 経験的知識を正当化する方法とその確認

(C02)正当化理由

ルールの根拠として、今までに整理されているブレイクアウト発生メカニズムを調査した。

(C05)過去のデータからの検討

過去の測定データから検討してみて、ルールの正当性を裏付けるブレイクアウト発生メカニズムからは、ルール条件部の属性に対する裏付けは取れるが、属性値の裏付けが取れるほど詳細なものではないことが判明した。

(1-5) 知識表現の選択と適切さの確認

(C07)知識表現の保守容易性、(C04)知識表現の理解容易性

ルールシンタックスは、実装技術とかなり結び付いた形で記述されているところがあるため、専門家にとって分かりにくくなっており、保守容易性にも悪影響を与え、改良の必要があるが、現在また行っていない。

(1-6) 知識獲得方法の選択と適切さの確認

(C03)知識の収集

ブレイクアウトがそれ程頻繁に起こる事象ではないことから、主知識源の同定が困難であり、データを主知識源として、ルール帰納型ツールやニューラルネットツールの利用も考えられた。しかしながら、数は少なくても、経験的知識が存在するので、やはりそれを主知識源として、データは知識の洗練化作業時に利用することにした。

(C12)テストケースの収集

本プロジェクトでは、システムの開発と並行して、データも収集し、感度解析に有効と思われるデータが収集される度に、専門家に参加してもらって、知識ベースの妥当性の確認を行った。

(2) 問題解決・推論方式の妥当性評価

(2-1) 推論処理の妥当性評価

(C01)試行錯誤的処理

本システムは、基本的に、測定データのある特徴的なパターンを認識してプラントの異常状態を診断するものであり、試行錯誤的な要素はほとんとない。しかしながら、その特徴的なパタン自身が整理されておらず、過去のデータを通して、有効なパターンを把握していくプロセスに、試行錯誤的処理の要素が含まれているため、エキスパートシステムの開発方法論を採用した。

(C03)曖昧性の取扱い

ルール及び記号自身の持つ曖昧性については、ファジィ理論等の応用が考えられたが、ルールを構成するプリミティブ(記号)自身の洗いたしや最適化が未了である時点で、本項目を考えるのは早いと考え、導入を現時点ては見送ることにした。

(*C1)データの利用

数は少なくても、経験的知識が存在することから、データを主知識源とはしないで、知識の洗練化作業時に利用することにした。

(*C2)対象モデルの利用

ルールの根拠（深い知識）としては、ブレイクアウト発生メカニスムが考えられたが、ルールが生成てきる程の「詳細さ」は備えていないため、ルールチェックの利用にとどめた。

(2-2) 推論時間の妥当性評価

(M02)推論エンジンの処理時間

Prologで開発した推論エンジンでは、ブレイクアウトの認識に２０秒以上必要とし、処理時間に関する要求事項を充すことができなかった。そこて、プリミティブなレベルで処理時間を測定した結果、当初、「単一化に多くの時間が裂かれているのだろう」という予想に反して、コンバータによって変換された記号データをコンサルトする時間がほとんどであった。そこで、エキスパートンステムの実行部については、プロトタイプシステムと実用システムの２種類を用意することにし、後者はＣ言語で開発することにした。

(2-3) 推論手法の選択と適切さの確認

(2-4) 不確実情報に対する処理技法の適切さの確認

(C04)不確実性の意味

記号自身については、時系列数値データの変化傾向（トレント）を２データ間の角度によって、７種類の記号データに対応付けているが、その対応付け方に不確実性が発生している。また、そのトレントを表す記号の属性として、持続時間などを考え、その値がルールに記述されているが、それらの属性および属性値の記述にも不確実性が発生している。

(2-5) 学習機能の適切さの確認

(C02)学習機能利用の利点と問題点

本エキスパートシステムの開発において、利用可能な知識源とその特性は以下の通りである。

知　識　源	特　　性
対象モデル	かなり定性的なレヘルである。
経験的知識	曖昧ではあるが、幾つか存在する。
データ	多量に存在する。

対象モデルが十分であれば、演繹的学習の利用が考えられたが、本エキスパートシステムの場合は、かなり定性的で詳細度が低かったため、利用しなかった。

また、データが多量に存在するので、ルールインダクンョンあるいはコネクンョニズムの利用が考えられたが、入力データの記述子がこれらの帰納的学習には大きく影響し、安易に導入すれば、最初に与えたルールの条件部によって形成される（ブレイクアウトの）バージョン空間内での安易な最適結果のみを出して終わることに成りかねない。

そこで、曖昧であるが幾つか存在する経験的知識をバージョン空間の初期地点として、新しくバージョン空間へも移行できる様な知識洗練化機構を開発することにした。

しかしながら、この知識洗練化機構においては、与える学習テータの順序が洗

練化の偏りを生んでおり、この問題を解決しなければならない。また、対象モデルの深い考察から、新規属性を増加させる必要がある。

(M01)学習前／後の性能比較

図4 2-1 の枠組みにより、今までに採取されたデータを学習データとし、知識洗練化の評価基準（認識率）を「ブレイクアウトの網羅率 － 正常データのブレイクアウトへの誤認率」として、実験を行った。6個の初期ルールから始めた場合の洗練化実験の結果を図4.2-2 に示す。

この結果から、属性値の最適化を試みる一般化・特殊化戦略の適用だけでは知識ベースの洗練化には限界があり、タスク戦略による新しい属性の付加が必要であることが判明した。

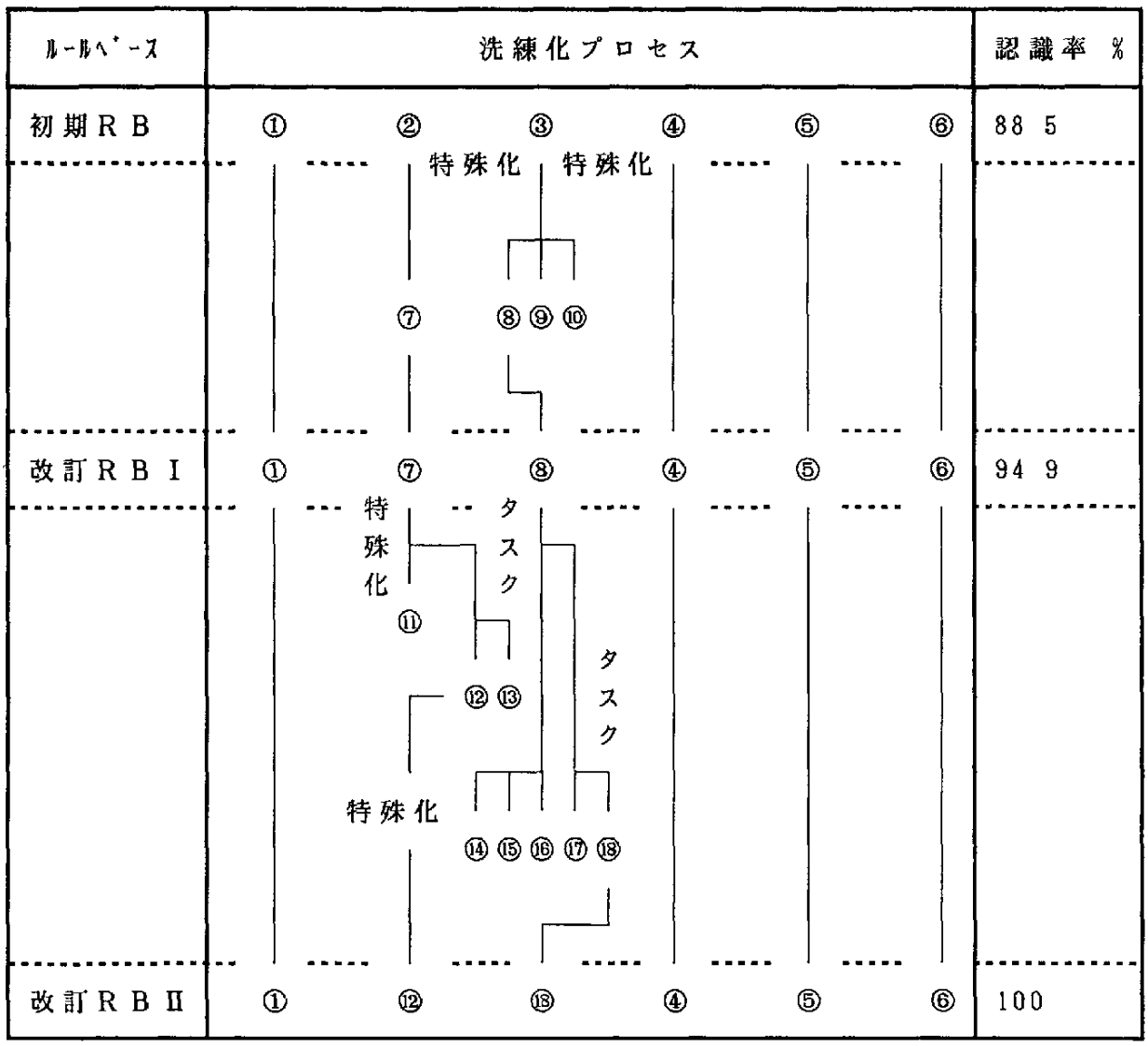


図4 2-2 ルール洗練化実験の結果

(3) 結果の妥当性評価

(3-1) 正解の評価

(C01)正解に関するコンセンサス

「現実のプラント状態」と「エキスパートシステムの認識結果」の一致が「正解」であり、不一致が「不正解」である。

しかしながら、不正解における以下の２つのケースは、実際には、意味合いが大きく変わる。

	ﾌﾟﾗﾝﾄ状態	ｴｷｽﾊﾟｰﾄｼｽﾃﾑ	対処方法
①	正常	ﾌﾞﾚｲｸｱｳﾄ	ﾌﾟﾗﾝﾄを止めて、ﾁｪｯｸし、再作動する
②	ﾌﾞﾚｲｸｱｳﾄ	正常	ﾌﾞﾚｲｸｱｳﾄが起こり、ﾌﾟﾗﾝﾄが暫く使用不可能

即ち、①より②の方が、起こっては困るケースである。

そこで、単純に正解率でシステムを評価するのではなく、②をブレイクアウトデータの検出率、①をブレイクアウトへの誤認率として、②－①でシステムを評価することにした（便宜上、ブレイクアウトの認識率と呼ぶ）。

(C03)限界値テスト

実験を進めていくにつれて、限界値データの性格が把握できたが、そのようなデータを人手で作成してもほとんど意味がなく、実際のデータを利用することはできなかった。

(C04)オンラインデータ

エキスパートシステムのテストには、オンラインで収集された実際のデータを使用した。

(M01)正解率の測定

認識率は、88 5% → 94.9% → 100%と改善されていったが、未知のブレイクアウトに対してどれだけの検出率があるかは当然不明であり、この点に関する何らかの指針の設定が課題となった。

(3-2) 解の網羅性評価

(3-3) 解の最適性評価

(3-4) 解の評価方法の適切さの確認

(C01)評価方法

過去に測定した実際のデータにより、ルールの形式を一時的に固定して、実システムをＣ言語で開発し、フィールドテストにより、不都合が生したら、知識洗練化部を再び稼働して、ルールを再洗練化することにした。

(C02)解の性格

正常データの方が圧倒的にブレイクアウトデータより多いため、(3-1) の評価基準では、その値がたとえ９０数％でも①のケースは頻繁に起こっており、甘い評価基準と言わざるを得ない。

しかしながら、②のケースが全く起こらないことを第一目標として、本評価基準を先ず採用した。

(4) 要求仕様に対する達成度評価

(4-1) 要求仕様内容に対する評価

(4-2) 要求仕様に対する達成度評価

(C01)(C02)第Ⅰ／Ⅱ版要求仕様書

ンステムの目的は、ブレイクアウトを検知し、未然にブレイクアウトを防くことであり、それ以上、詳細な事項は決定されなかった。

機能要件としては、入力は3種類のセンサーからのサンプリングデータであり、処理は10数秒以内にこれらのデータを解釈してブレイクアウトを認識することであり、出力はプラント操作員を支援する形で利用されることが決定されたたけであった。

(C03)実用版要求仕様書

発展型プロトタイプの構築時に、実用版要求仕様書が作成された。実用システム稼動時の利用計算機のハードウェア仕様が決められると共に、計測データの伝送仕様が決められ、エキスパートシステムの応答時間の上限が設定された。

また、操業条件に関する処理がブレイクアウトの誤認率を下げることに大きく貢献することが判明したが、その表現方法が問題となり、当面、鋳造速度と温度データの関連性により操業条件を表現することで対処することにした。

(C04)計画と実際の開発とのギャップ

本エキスパートンステムは、現存のルールベースにより対処できないデータが出てきたら、その状況に対処てきるようにルールを改訂する知識洗練化部を持つことを特色としているが、上記の操業条件に対しては考察が進んでおらす、課題となっている。

(4-3) 要求仕様設定手段の確認

(5) 知識ベースの保守性評価

(5-1) 変更箇所発見の容易性評価

(5-2) 知識ベースの理解性評価

(C04)記述レベル

現存のルールの記述レベルは実装技術と関連する所が多分にあり、専門家にとって分かりにくくなっている。従って、より分かりやすい上位のルール表現を用意して、ルールに変換するトランスレータの開発が必要である。

(C06)技術移転

システム開発の重要な点については専門家に随時伝達し、ルールンンタ｡クスなど専門家にとっても関わりの深い所は、なるへく理解してもらう様に心掛けた。

(5-3) 変更箇所の局所性評価

(5-4) 知識ベースの統一性評価

(5-5) 品質安定性の確認方法の確認
(C03)バージョン管理
ルールの洗練化に関するバージョン管理については、ルールの記述子の中に親ルールのＩＤを持たせると共に、洗練化プロセスを木で表示できるようにした。

(5-6) 保守マニュアルの整備状況の確認

(5-7) 保守用ツールの整備状況の確認

(6) ヒューマンインタフェースの妥当性評価

(6-1) 利用時のインタフェース評価
(C01)結果の表示
ブレイクアウトを認識した時は、いち早く人間に知らせるために、ブザーを鳴らすと共に、すぐにプラント状態を把握できるように、チャート形式てサンプリングデータを表示している。

(6-2) 保守時のインタフェース評価

(7) システムの総合評価

(7-1) テーマ選定の妥当性評価
(C01)ニーズの分析
システム化のメリットとしては、専門知識の整理が促進されることが挙げられる。従って、本システムは、最終的には専門家の作業代行を目指しているものの、運用を通して、システムの信頼性が向上するまで、当面は、専門家の作業支援として位置づけた。

(C04)試行錯誤性
知識を利用して推論するプロセスには、試行錯誤的な要素はほとんと含まれないが、知識を確定していくプロセスが試行錯誤的である。

(C07)過去のデータの利用
オンラインで収集された実データが利用可能である。

(C11) 100% の正解率
正解率が100%であることが当然望ましいが、現時点て専門知識があいまいてあることから、当面は、100%の正解率は要求されない。しかしながら、ブレイクアウト発生時には必ずそれを認識するという完備性については、100%要求される。

(7-2) 利用者の満足度評価

(C09)最小限のデータ

ブレイクアウトがまれな現象であるため、その現象に関するデータの獲得が難しく、結局、数年程度のデータ収集は必要であると思われる。

(C10)出力結果の視覚化

サンプリングデータはチャート表示し、専門家が出力結果を考察し易くしている。

(C13)学習機能

本システムでは、知識洗練化機構を備えることにより、ルールベースが自動的に更新されることて、システムの逐次的な成長を自動化している。

(7-3) 統合性の評価

(C02)他システムとの連続性

実用システムにおけるデータ伝送は、工業用ＬＡＮとして普及しているＭＡＰに準拠したＬＡＮシステムを採用し、そのネットワークトライバを通して、データの授受が行われている。

(7-4) 成長可能性の評価

(C04)期待度／失望度

ブレイクアウト発生時には必ずそれを認識するという完備性については、100％を達成することは可能であるが、正常時をブレイクアウトに誤認する率を０％にすることは極めて困難であることが予想され、エキスパートンステムの利用方法についても考慮しなければならないかもしれないという意見が出た。

(C06)開発環境と実行環境

①頻繁にルールを変更することと、②処理時間の厳しさの２つの要求から、エキスパートンステムの実行部については、①を充すプロトタイプンステムと②を充す実用システムの２種類を用意し、前者はProlog言語、後者はＣ言語により開発した。

(7-5) 開発／保守用トキュメントの評価

(7-6) 開発方法の適切さの確認

(C05)実績

以前に開発したンステムで使用した、株価テクニカル分析ルールの洗練化と同様のアプローチを採用し、初期に与えたルールの改良にある程度の成功を収めている。

(7-7) ツール・言語選択の適切さの確認

(C07)ツールの制約

Prolog処理系にコンサルトてきるデータ量には上限があるため、サンプリングデータを一度にどの程度処理するのかについて、検討する必要があった。

(7-8) システム開発の生産性評価

(7-9) ンステムのコストパフォーマンス評価

(7-10)セキュリティ対策の評価

4.3 ＬＳＩ論理合成エキスパートシステムへの適用

4 3.1 概　要

本節では、ＬＳＩ論理合成エキスパートシステム(EXLOG) のプロトタイプ版を改善して実用版（Ｖ１）を完成させる過程［岩本　1989］で、本カイトラインの内容が如何に有用であるかを示す。つまり、EXLOG プロトタイプ版に対して本ガイドラインの内容を適用することで、そのプロトタイプ版の問題点が抽出され改善項目が明確になり、EXLOG 実用版（Ｖ１）に到達できることを紹介する。なお、EXLOG 実用版の実際の開発が、本ガイトラインに沿って行われたわけてはなく、EXLOG 実用版の開発経験が１つの事例として活かされ、本ガイトラインが編成されたことをお断りしておく。

4 3 2 ＬＳＩ論理合成エキスパートンステム(EXLOG)とは

以下に、EXLOG の入出力、設計対象モデル、及び処理の概要を述へる。

EXLOG の入力はレジスタトランスファレベル記述言語で書かれた要求仕様、出力はゲートスイッチレベル記述言語で書かれた回路記述、及び回路図である。EXLOG では、要求仕様、出力回路記述を含む全ての設計対象をノートパスモデルで表現している。ノードパスモデルとは、ノートとパスからなるオブジェクトのネットワークである。ノートは入力信号に特定の操作を加えて出力する実体て、図4.3-1 におけるnodeのインスタンスとして表現される。パスは信号を伝播する実体で、pathのインスタンスとして表現される。

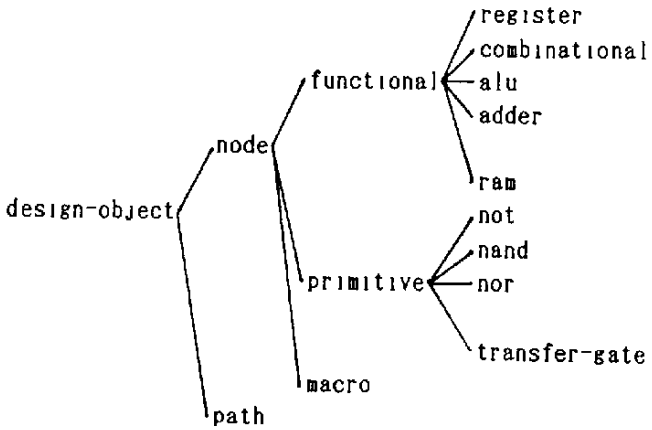


図4.3-1　設計対象のクラス階層

EXLOG における論理合成過程を、ノートパスモデル上で述へる。まず、与えられた要求仕様記述は、functionalのインスタンスとパスからなるネットワークに変換される。各functionalのインスタンスでは、実現すへき機能仕様が示されている。なお、論理合成の中間状態にあるノートもfunctionalのインスタンスとして表現される。続いて、詳細化、ブール式レベルでの最適化、テクノロジーマッ

ピング等の操作がノードパスモデル上でなされる。詳細化とはノートをより具体化されたノード群に分解する処理であり、例えばregisterのインスタンスの機能仕様中の、加算演算を行っている部分を加算器(adderのインスタンス)として分解する処理である(図4.3-2)。また、ブール式レベルての最適化は、combinationalのインスタンスに示されているブール式(仕様)を、より最適な記述に書き換える処理である。また、テクノロジー変換はテクノロジーに依存した回路素子(primitiveのインスタンス)、及びマクロ(macroのインスタンス)へマッピングする処理である。ここでマクロとは、ある機能を実現する構造が与えられているオブジェクトである。

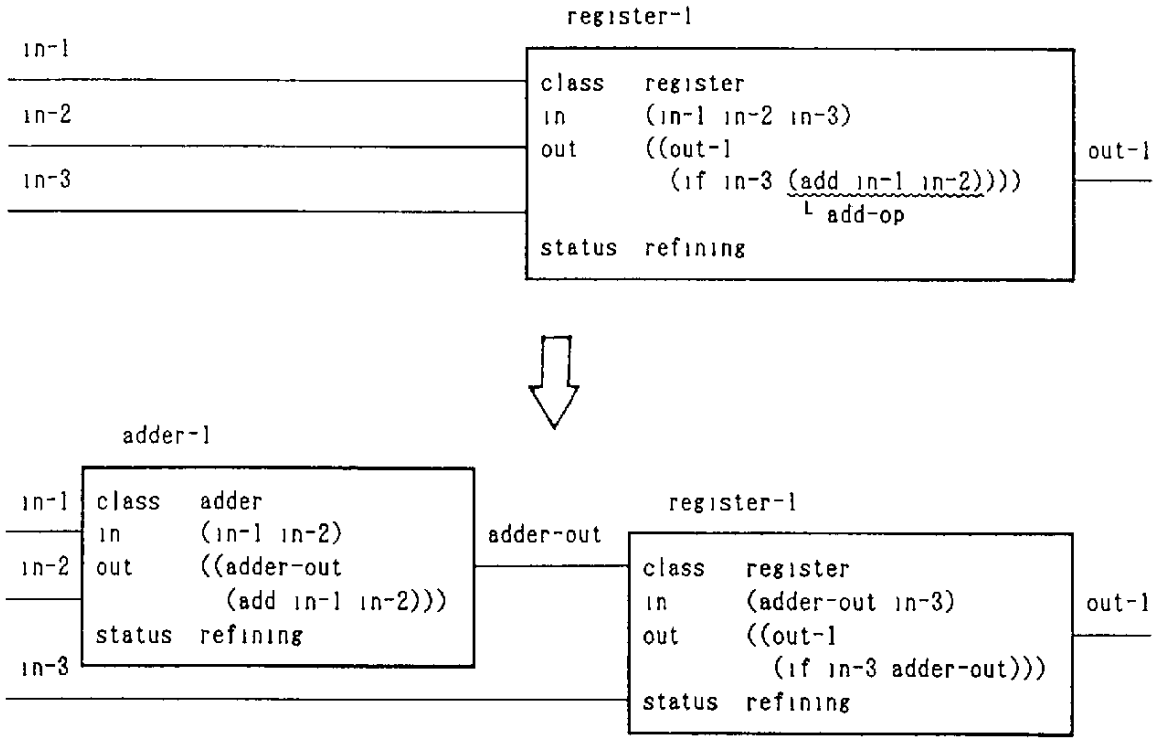


図4.3-2 ノート分解

EXLOG では、以上の論路合成処理のほとんどを、ルールベース処理として実現した。その理由は、全てが明かになっていない設計ノウハウを変更追加していく上で、ルールベース手法により段階的に行わざるをえなかったからである。

4.3.3 本ガイドラインによるEXLOGプロトタイプの評価

EXLOG プロトタイプ版に本ガイトラインの内容を適用して、EXLOG プロトタイプ版の問題点が明確化されることを示す.

4.3 3 1 プロトタイプ版での問題点

本ガイドラインの内容を基に、プロトタイプ版を評価した結果、以下の問題点が明らかになった。

[問題1] ルール記述性の問題

←: [本ガイドライン]: 1－1 知識ベースの正当性評価
1－5 知識表現の選択と適切さの確認

各ルールのRHS(right-hand-side)で行う処理が複雑であり、RHSを書くのが容易ではない、また、何を行っているかが一目では分かりにくい、という問題てある。プロトタイプでの問題点を具体的に明示するために、図4 3-2 に示すノートの分解処理を考える。つまり、「あるレジスタ(register-1)の要求仕様中の加算演算を実現する部分((add in-1 in-2))を、独立した加算器(adder-1)として取り出す」という処理を行うルールにおけるRHSの実現方法を考える。プロトタイプては次のレベルでルールのRHSを記述した(図4 3-3(a))。

(1) ノードadder(adder-1)を作る。
(2) adderの出力をreg(register-1)に接続する。
(3) adderの仕様をadd-op((add in-1 in-2))とする。
(4) add-opで参照しているパスをadderの入力として接続する。
(5) regの仕様中のadd-op部分をadderの出力で置き換える。
(6) regで不用になった入力をregから切り離す。

但し、このルールはプロトタイプで実際に使用したものを簡略化している。なお、LHS(left-hand-side) では、クラスregisterのインスタンスて、statusスロットの値がrefiningで、かつ、仕様中にadd で始まるパターンが含まれているものがあるかどうかを調へている。

この例で示したようにプロトタイプのRHSは、ノートパスモデルの個々の構成要素(ノート、パス)の生成/削除/書換、等の操作レベル(以下、このレベルをデータ操作レベルという)の記述てあった。従って、同しノートの分解処理を行っているルールでもルール作成者毎に様々な表現が可能となり、作成者以外はルールの内容を理解しづらい。当然のことながら、このレベルの記述では記述量も多くなり、中には100 行になるルールもあった。

```
(defrule r-decompose-adder
  (there-exists ?reg (ii* $register)
    (eq (the-status ~reg) 'refining)
    (match (specifications ~reg)
           {(@has (@and (add ? ?)
                        ?add-op)})))
  ->
  (bind ?adder (new-node $adder ~reg))
  (bind ?adder-out
        (new-path (width ~add-op)
                  ~adder ~reg 'data))
  (put-spec ~adder ~adder-out ~add-op)
  (dolist (path (in-paths ~adder))
          (connect path ~adder 'data))
  (bind ?before-in-paths (in-paths ~reg))
  (subst-spec ~adder-out ~add-op ~reg)
  (dolist (path (path-diff ~before-in-path
                           (in-paths ~reg)))
          (disconnect path ~reg)))
```

(a) A Rule from The EXLOG Prototype

```
(defrule r-decompose-adder
  <no difference from EXLOG prototype on LHS>
  ->
  (decompose ~reg ~add-op $adder))
```

(b) A Rule from The EXLOG V1

?..: variable ~..: value of variable

図4.3-3　EXLOGプロトタイプとV1とのルール比較

［問題2］：設計品質の問題

←：［本ガイドライン］：３－３　解の最適性評価

プロトタイプでの設計結果を熟練設計者により設計されたものと比較した結果、トランジスタ数で30% 増となり、実用上、問題となった。また、論理設計後のレイアウトを考慮した最適化という点でも問題があった。例えば設計者は、必要に応じて設計途中でマクロを定義し、部分回路の繰り返し構造を明示することにより、設計品質を向上させている。プロトタイプの場合は、既存のマクロを利用する以外は、全てゲートスイッチレベルまで個々に展開した。このため、同じ構造を繰り返し使うべき箇所も、個々独立に合成された。

［問題3］：処理速度の問題

←：［本ガイドライン］：２－２　推論時間の妥当性評価

一般に、ルールベースに基づくシステムの処理は、従来の手続き的な手法に基づくものに比較して遅い。プロトタイプでは、明らかにルールで処理する必要がない部分(入力FDL記述の構文解析、出力GDL記述の生成等)は従来言語(C 等)を使用して手続き処理で実現した。しかし、プロトタイプては約1000ゲート規模の回路の合成に、4MIPS 相当のワークステーションで約45分を要した。

4.3.4　ＥＸＬＯＧ実用版への発展

プロトタイプ版の評価結果に基づいて如何に改良して実用版としたかを示す。つまり、以上に述べた問題点を解決するためには、フルカスタム設計に必要な知識を簡潔に記述するための枠組みと高速実行のための制御方法の検討が重要となる。以下、これらの問題点を解決するためにEXLOG 実用版が採用した設計過程モデルとその制御について述べる。

4.3.4.1　設計過程モデルとその実現

設計過程モデルは、設計操作を適切なレベルで認識、実現するための基盤となるものである。提案する設計過程モデルは、フルカスタム設計に必要十分な設計操作単位(作用素)から構成される。システムをルールベース手法により実現する場合、モデルを構成する作用素に、RHS 関数が対応する。

設計過程モデルのポイントを次の２点とした。

①フルカスタムLSI 設計で必要とされる高度な操作を実現する作用素のセットとすること。

②個々の作用素を適切なレベルで抽象化し、わかりやすいものとすること。

作用素の決定は、プロトタイプでの全ルールの分析と、実用版て新たに実現すべき機能の洗い出しに基づいて行った。その結果、16個の作用素を抽出し、基本的にルールのRHS で使用可能な関数として実現した。

（１）設計作用素

表4 3-1 に16個の作用素（関数）と、その処理内容、及び、用途を示す。16個の作用素はノート内作用、つまりノート内に閉じた処理をするものと、ノートパス作用、つまりノートとパスのネットワークを変更する処理をするものから成る。さらに、ノートパス作用素はその変更のタイプによって、分解系、併合系、置換系、追加削除系に分類される。分解系は１つノートを複数のノートに分解するもの、併合系は複数のノート、パスを１つのノートに併合するもの、置換系は単（復）数のノートから単（復）数のノート（primitive、macroのインスタンス）への置換を行うもの、そして、追加削除系はノートの挿入、削除を行うものである。

作用素は、表4.3-1 に示す論理設計の各フェーズで使われている。

以下、作用素により実現された機能の中で、フルカスタムLSI 設計て重要となるものを示す。機能レベル最適化機能はフルカスタム設計固有のものてはないが、厳しい要求性能を満足するための最適設計には不可欠なものである。また、マクロの動的定義機能、ダイナミック回路による合成機能は、フルカスタム特有の設計を行う上で必要となるものである。それぞれ、レイアウトを意識した設計、トランジスタの特性を利用した設計を行う上で必要となる。

(1-1) 機能レベル最適化

複数ノートのグループ化、不用ノートの削除、機能仕様の書き換え等の機能レベルの最適化が、fold、change-class、delete-node、rewrite等の設計作用素を使って実現された。複数ノードのグループ化の例としてALU 認識の過程を挙げる。ALU の認識は組合せ論理埋め込みルールとALU 判定ルールとによってなされる。組合せ論理埋め込みルールでは、「ある論理を実現しているクラス combinationalのインスタンス(node)の出力先が１つだけ(dst)で、かつdstがクラスregisterのインスタンスでnodeからの出力をデータとして(コントロールてはない)利用している時に、nodeをdst に埋め込む[fold]」ことである。

この処理において、node内で実現されている論理が、加算、減算等の算術機能論理演算であった時には、加算器、減算器のALU へのまとめ上げの一過程と捉えることができる。ALU 判定ルールは、そのまとめ上げられたノートに対して適用される。このルールは「クラスregisterのインスタンスで、その仕様の形式は排他的条件文（同時に複数の条件が成立しない）であり、各作用部で算術機能論理演算を行っており、その種類が一定数以上であり、データ入力がラッチからの出力である時に、nodeのクラスをalu に変更し[change-class]する」ことである。

なお、排他的条件とその条件下での作用はcond文で記述する(記述順序に依存するlispのcondとは異なるセマンティクスを持つ)。各機能を個別に合成する場合と比較して、かなりコンパクトになる。

不用ノードの削除を行うものとしては、入力仕様を記述する言語FDLにおいて、ある特定ビットのみが変化するレジスタを１文だけでは記述することがてきないことから生したダミーのレジスタを削除するためのルールがある。機能仕様の書き換えを行うものとしては、コントロール入力をデータ入力に変更することにより、より単純なレジスタに変換するルールがある。例えば、“ if x*y then 1 else if (not x)*y then 0 ”という仕様記述を“ if y then x ”に書き換える。それぞれ、作用素delete-node、rewriteが使われている。

表4 3-1　設計作用素

		作用素	処理内容	用途				
				機	詳	プ	テ	ゲ
ノード内作用		(rewrite node old-spec new-spec)	nodeのold-spec部分をnew-specに書換え	○	○	○		
		(change-class node class)	nodeのクラスをclassに変更	○				
ノードパス作用	分解	(decompose node sub-spec class)	nodeのsub-spec部分を分解		○		○	
		(decpmpose-bitwise node class)	nodeをビット単位に分解		○		○	
		(decompose-internal node pin class)	nodeの内部ピンpinの部分を分解				○	
	併合	(fold node)	nodeをその出力先に埋込み	○				
		(merge-sequential src dst class)	入出力関係にあるノートsrcとdstを併合			○		
		(merge-self node path)	nodeへのフィードバックpathの取込み	○				
	置換	(instantiate-primitive node prmtv)	nodeをprmtvのインスタンスと置換				○	
		(instantiate-macro node macro in-list out-list)	nodeをmacroのインスタンスと置換				○	
		(instantiate-auto-defined-macro node)	1ビット分のﾏｸﾛを定義し、nodeの前後をそのﾏｸﾛのｲﾝｽﾀﾝｽと置換	○				
ノートパス作用	置換	(transfer-to-circuit node circuit)	nodeをcircuitのインスタンス群と置換				○	
		(share original identical)	identicalをoriginalで置換、共有					○
	追加削除	(insert-node class)	node下流にclassのインスタンスを挿入					○
		(delete-node node)	nodeを削除	○				○
		(add-signal inserted-path original-path node)	original-pathとnodeの間にorを挿入、orの入力にinserted-pathを付加					○

機･機能ﾚﾍﾞﾙ最適化　詳 詳細化　ﾌﾟ ﾌﾟｰﾙ式ﾚﾍﾞﾙ最適化　テ テクノロジ－マッピング
ゲ ゲ－トﾚﾍﾞﾙ最適化

(1-2) マクロの動的定義

作用素instantiate-auto-defined-macroによって行う。この作用素は、図4 3-4に示すようにいくつかのノードが多ビット接続している時に適用される。この図の場合、8ビット幅でレジスタA、B、Cが接続されている。各レジスタでの演算が論理積、論理和等ビット毎に閉じた（加算、シフト等、上位下位ビットに影響しない）演算であれば、各ビットとも同じ構造で実現することができる。4 3 3 1でも述べたように、人手で設計する場合、繰り返し構造をできるだけ多く見つけ、作ろうとする。なぜならば、規則性が高い方が、合成結果が分かり易く、また次の過程でのレイアウトも行い易いからである。

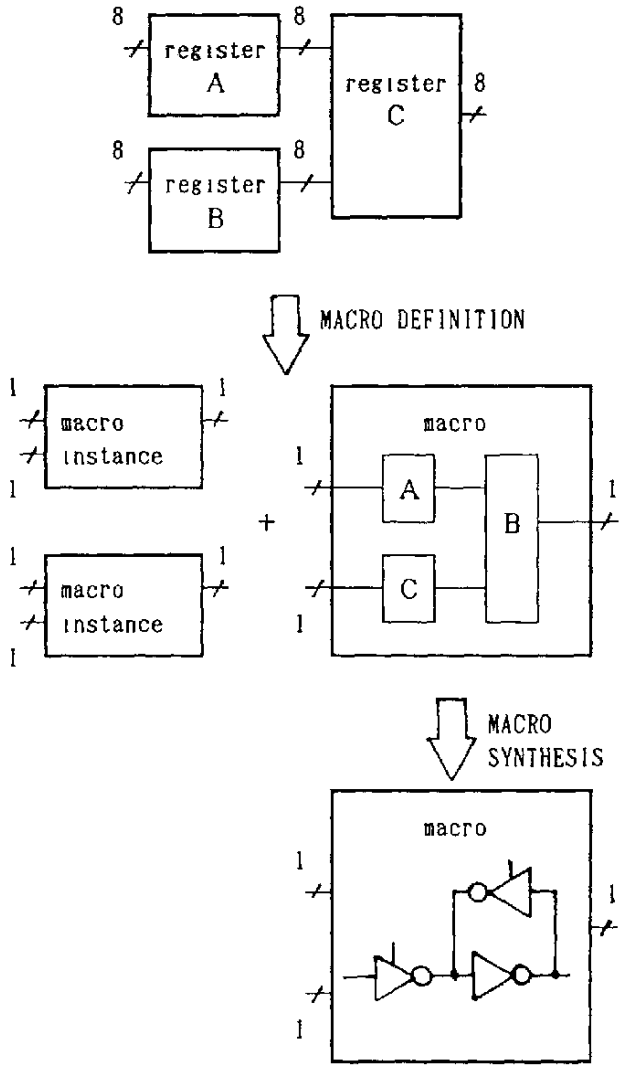


図4 3-4　マクロの動的定義

EXLOG でも同様のことを行う。つまり、繰り返し単位として1ビット分のマクロオブジェクトを定義し、元のノートはそのマクロを参照する形（マクロのインスタンス）に置き換える。図4 3-4の例では、レジスタA、B、Cで行っていたのと同じ演算を行う1ビット分のマクロを定義している。そして、8ビット幅データ

を扱うレジスタA、B、C は定義されたマクロを参照するノート８個と置き換えられる。以上の過程をマクロの(動的)定義と呼び、作用素instantiate-auto-defined-macroによってなされる。

定義されたマクロはまだ、ゲートスイッチレベルにまでは詳細化されていない。定義されたマクロも１つの独立した回路とみなされ、論理合成処理が再帰的に適用される。この過程は設計対象に対するものと同一であるが、特に区別してマクロ合成という。

(1-3) ダイナミック回路による合成

ダイナミック回路を用いる際には、そのプリチャージ、チャージオフ等の制御のタイミングを決めなければならない。作用素instantiate-macro は、利用するマクロがダイナミック回路であった時に、マクロ中に指定された動作条件（入出力信号と制御信号とに関する制約）と、マクロのデータ入出力に接続されるパスの有効タイミングとから、マクロの適切な制御信号を生成する。

（２）作用素の抽象度

本稿で提案した作用素は、論理正当性保存変換(correctness-preserving transformation) を基本とする設計操作レベルのものと言える。論理正当性保存変換とは、ある部分回路を論理的に等価な別の回路に変換する操作である。プロトタイプにおけるルールの記述は論理正当性保存変換でないデータ操作レベルでの記述であったため、ルール作成者のミスにより、論理的に不整合な状態になることが多かった。一方、実用版V1では不整合になる可能性は減少している。各作用素の中で、分解系、併合系のものに関しては、不整合になる可能性はない。また、ノード内作用素、及び置換系のものに関しては、引数の与え方によっては不整合となりうるが、変換前後の論理的等価性のチェックは容易であり、不整合状態の発生の可能性は低い。追加削除系の作用素は論理正当性保存変換をするものではない。これらはバッファ(インバータ)の挿入、削除に使われることか多いが、ほとんどの場合ペアで使われ、論理的不整合を生しる恐れは少ない。

また、プロトタイプでの分解ルール(図4 3-3(a))と実用版V1でのルール(図4 3-3(b))を比較すると、decomposeといった設計者にもなじみやすい概念レヘル(設計操作レベル)に抽象化されていることがわかる。

4.3.4 2　設計作用素の最適制御

知識の記述性を保ちながら、合成処理を高速化するために、作用素の最適制御構造の検討を行う。各作用素は、その種類と使用目的とにより、とう制御すへきかということが決定される。作用素の起動条件が複雑な場合には、作用素をルールのRHS として利用する。そうでない場合は効率上、手続き的処理の一部として利用するのがよい。以下、(1) 作用素をRHS として実現した場合のルールの制御の最適化、(2) 手続き的処理の一部として使われる作用素と、手続き的処理とルールベース処理との併用について述へる。

（１）ルール制御の最適化

作用素をルールのRHSで使用する場合、作用素の制御はルールの制御に当たる。ここでは、ルール制御の最適化を高速化の観点から行う。

作用素の種類とその使用目的によって、ルールの照合処理を省略できるものがある。例えば、図4 3-5に示すルールセット（ルールの集合）rs-decompose-special-functions による合成処理は、作用素decompose を用いて、算術演算、機能演算を行っている部分の分解を行っている。この場合、各ルールの照合は、一度失敗すると二度と成功しない（他のルールの適用によって、照合が成功する状況になることはない）。以上のような処理を行っている部分については、逐次的にルールを適用していき、最後のルールの適用後終了する制御構造が適している。以上の制御を行う方式(fire- until-fail-once) を定義し、不用なルール照合を省き、高速化した。

```
ruleset rs-decompose-special-functions
  normal-rules.  (r-decompose-alu
                  r-decompose-comparator
                  r-decompose-adder
                  r-decompose-subtracter
                  r-decompose-multiplier
                  r-decompose-carry
                  r-decompose-barrel-shifter
                  r-decompose-barrel-rotator
                  r-decompose-timer
                  r-decompose-counter-up-down
                  r-decompose-counter-up
                  r-decompose-counter-down
  control-strategy: fire-until-fail-once
```

図4.3-5　ルールセット例

（２）ルールベース手法と手続き的手法の併用

起動条件が単純な作用素については、特に複雑な制御構造にせず、手続き処理の一部として利用するのが効率上適している。ここては、ルールベース手法と手続き的手法を如何に切り分けたかを述べる。基本的に、ルールベース手法の中に、手続き的手法を取り込む形で両手法を併用した。

ルールベース手法の制御上の長所は、(1) 処理単位(ルール)の独立性（制御の分離）、(2) パターンマッチングによる起動、短所は(3) 処理効率の悪さ、である。一方、手続き的手法は、ルールベース手法の長所は損なわれるものの、処理効率は期待できる。そこで、ルールベース手法の長所を必要としない作用素の適用処理、つまり、あるまとまった一連の操作を、パターンマッチング等の複雑な条件によることなく行う合成処理を、効率のよい手続き的手法で実現した。

手続き的手法で実現した合成処理を２例示す。一つは、作用素merge-sequentialを用いて、組み合わせ論理を表している２ノートの併合処理を行うものである。これは、ブール式レベルでのファクタリング処理等を効率良く行うための処理てある。この併合処理の条件は、両ノートが共にクラスcombinational のインスタンスであること、かつ両ノートがパスを通じて直接接続されていることてあり、複雑なパターンマッチングは必要としない。

もう一つの例は、combinational のインスタンスの機能仕様内てのブール式レベル最適化処理の終了後に、作用素decompose、decompose-internal、instantiate-primitive等を用いて、ノートをprimitive オブジェクトのインスタンス集合へ展開する処理である。この場合の条件はノートのクラスがcombinational てあることのみである。

なお、ルール制御方式の定義には、EXLOG で利用したツールCLのルール制御戦略を宣言的に定義する機能(トメイン言語プリミティブ)を利用した。また、同しく手続き処理との併用には、処理単位(タスク)として、ルールセットに基づくものとC、Lispで記述されたものを指定できる機能を利用した。

4 3.4.3 ＥＸＬＯＧ実用版の評価

ＥＸＬＯＧ実用版で採用した設計過程モデルの有効性を、4 3 3 1で述へた問題が如何に解決されたかの観点から評価する。

（１）ルール記述性

(1-1) 分かり易さ

ルールの記述量をプロトタイプと比較した。記述量の比較は各バージョンての全ルール(プロトタイプ 81ルール、V1 160ルール)のRHSの行数の平均によって行った。プロトタイプでは16 8行、V1では4 2行で、1/4となった。記述量の減少、及び、設計操作レベルへの抽象化により、各ルールの行っている操作は分かりやすいものとなった。

(1-2) ルール作成期間

ルールを記述するのに要した期間は、２度目のインプリメントということもあり、厳密には評価できないが、その影響を引いて、1/2 程度に短縮された。

(1-3) 設計過程モデルの妥当性

16個の作用素の中で、手続き的処理部のみで使われている３要素(merge- sequential、decompose-internal、instantiate-primitive)を除き、CL組み込みの２関数(bind.変数束縛、register-task.タスク登録)を加えた15関数てV1の全ルール(160ルール)をカバーした。但し、実用版以降の拡張の過程で追加されてたルールの中には、対応できないものも発生した。例えば、メモリ合成において、仕様中に明示されていないアトレスレジスタを分解するルールてある。現在、このルールは作用素とデータ操作レベルの関数とで記述されている。今後、作用素の拡張、改良の検討の必要性がある。

（２）設計品質

フルカスタム設計に必要な高度な作用素のセットを揃えたことにより、設計者のノウハウを容易にルール化することができた(ダイナミック回路による生成を行うルール等)。この他、機能レベルを含む各レベルでの最適化ルールを実現したこと等により、回路規模の点に関しては人手設計とほぼ同等の結果を得た(表4 3-2)。なお、ALU/PSWが、順序回路と組み合わせ回路とをバランスした回路であるのに対し、FIFOは順序回路部の占める割合が多い回路である。また、マクロの動的定義機能により、人手設計時の特徴である繰り返し構造の明確化が行なわれた。

表4 3-2　人手による設計とのトランジスタ数の比較

回　路	ALU/PSW	FIFO-1	FIFO-2
人手による設計	1832	994	1887
EXLOG プロトタイプ	2327 (127%)	−	−
EXLOG V 1	2156 (118%)	823 (83%)	1561 (83%)

（３）処理速度

手続き処理との併用により、ルールの総発火回数は約40% 減少した。なお、ルール数自体は最適化ルールを主体にプロトタイプと比較して約2倍(160ルール)に増えている。

手続き処理との併用、ルール制御の最適化により。処理速度は約2倍となった。なお、ツールCLの性能向上(ルールコンパイラ、オブジェクトアクセスの高速化)もあり、プロトタイプで4MIPS程度のワークステーションで、1000ゲート当たり45分を要していたものが、実用版では５分に短縮された。

また、マクロの動的定義機能も処理速度の向上に貢献している。つまり、前述したように、マクロ定義後は、１ビット分のマクロのみ合成すればよく、各ビット毎に同じ合成過程を繰り返す必要がないからである。

《参考文献》

[岩本　1989] 岩本、渡辺、河原林、前田．フルカスタムＬＳＩ論理合成エキスパートシステムＥＸＬＯＧにおける設計過程モデル、電子情報通信学会論文誌　Ａ，Vol J72-A, No 8, pp 1163-1171, 1989

4 4　水力鋼構造物評価診断システムへの適用

本節では、水力鋼構造物評価診断システム開発へのガイトラインの適用結果について述べる。このプロジェクトでは、診断対象は異なるが、エキスパートシステムとしては、機能・構成ともほぼ同一の２つのサブシステムを開発しており、ガイドライン評価の題材として適当と考えられる。

4 4 1　プロフィール

4 4 1 1　システムの目的と期待効果

我国で、発電用ダムおよび関連した構造物が建造されはしめたのは昭和初期である。それから60年近く経過した現在においても当時の構造物の多くが使用されている。（財）電力中央研究所では、古いダムについてその安全性調査を10年以上にわたって実施しており、すでに多くの経験・知識を蓄積している。そこで、昭和59年度より、これらの経験・知識を集大成し、総合的な安全性判断・寿命予測手法の確立をめざして研究を実施している。これまでに、現場での計測法の改善、全国共通の調査票作成と調査情報のデータベースシステム化、有限要素法による簡易解析システムの開発などを行なってきた。

本エキスパートシステムー水力鋼構造物評価診断システム（Hydro Steel Structure Diagnosing System, HSDS ）は、水力発電プラントにおける主要な鋼構造物であるダムゲート・水圧鉄管・水路橋・スクリーンの診断と寿命評価作業を、全国的な規模で同一の規準のもとに実施することを目指して開発された大規模な知識システムである。

ＨＳＤＳの目的は、従来の管理要領で「専門家に依頼する」としている部分の作業を明文化し、数多くのデータに基づいて、対象構造物の客観的・論理的な診断・評価を行いやすくすることである。そのために、ダムゲート診断システム(Dam Gate Diagnosing Advisor'DGDA)は、ダムゲートのデータベースから、変位・応力などの定量的な測定データ、ならびに外観状態などの定性的なデータを、また、簡易解析システムから、測定データに対応する解析値をとりこんで、個々の鋼構造物の総合的な診断・評価を行なう。そのため、データベース、簡易解析システムから得られる対象システムの構造・状態に関する情報と、ルールて表現した専門家の経験則を組み合わせて処理するのが特徴である[中村 1987]。

4 4 1 2　システムの概要

ＨＳＤＳには、知識ベースシステム、関係データベース、有限要素法による構造解析システム、統合的なユーザインタフェースから構成される。知識ベースシステムは、さらに、フレーム・ルールに基づく２つのサブシステム：ダムゲート診断システム(Dam Gate Diagnosing Advisor, DGDA)、水圧鉄管診断システム(Penstock Diagnosing Advisor,PSDA)にわかれる。関係データベースには、電力各社が所有する全国の水力鋼構造物の情報が含まれる。構造解析システムは、ダムゲート・水圧鉄管の典型的なモデルを保持しており、少数のパラメタを指定するだ

けで利用できる。ＨＳＤＳは（財）電力中央研究所の大型コンピュータ上て実現されている。利用者インタフェースは、各機能を統合して使えるように設計されて、端末、または、パーソナル・コンピュータ、ならびに、プリンタ・ファクシミリと接続することで全国的に利用てきる。

ＨＳＤＳの機能構成は、図4 4-1 に示すとおりである。システムはデータベース／簡易解析システムとともに利用される。利用者は、まず、現地での計測結果をもとに調査票を作成し、これをデータベースに投入する。次に、診断対象となる特定のダムゲートを検索・指定し、簡易解析システムを起動する。構造解析が終了すれば、解析結果とデータベース情報とを利用してエキスパートシステムによる診断・寿命評価を行う。エキスパートシステムに入力されるデータは、鋼構造物の構造形態・外観状態・操作状態・環境状態等の定性的情報と測定値・解析値等の定量的情報からなる。これらのデータはフレームに変換した後、当研究所の専門家の経験的な知識（ルール、手続き）に基づいて推論処理を行なう。これによって安全性判定ならびに余寿命予測を行なう。

知識ベースはフレームとルール、手続きとから構成される。フレームには上に述へた鋼構造物の定性的／定量的情報と対象システムの構成情報とを保持する。ルールは２つのグループに分れている。第１のグループは、入力したデータのチェックと各構成部分の評価スコアの計算とを行なうときに使用される。第２のグループは、さらに２種類に分類されており、外観状態と構造力学的観点とからダムゲートの診断・評価を行なうときに使用される。手続き的知識はフレーム中の個々の構成情報スロットに付随する関数であり、それぞれ評価スコアを計算するために使用される。

本システムの扱うデータは測定・解析による客観的な定量データとアンケートによる主観的な定性データの両方を含む。したがって、それらを用いて下す結論は本質的にあいまいなものとなる。本システムではここで生ずるあいまいさを処理するために、評価用スコアの導入と複数の解の提示という、２つの方式を採用している。これは次の理由による[中村 1986]。

①システムの下す複数の結論は互いに矛盾する可能性がある。ところが客観的な寿命評価手法が確定していないという現状のもとでは、システムを利用する専門家に、システムの結論の評価をまかせるへきである。そのためには、システムが結論し得るすへての解を表示することが適切である。

②鋼構造物のようなエンジニアリングシステムを対象とする場合、個々の推論結果は、データまで遡って明確に説明できるものでなければならない。それには、スコアの評価を陽に計算する方式が明解である。そしてこのスコアの妥当性を保証するために、ＡＨＰを使って定めた評価テーブルを採用している。

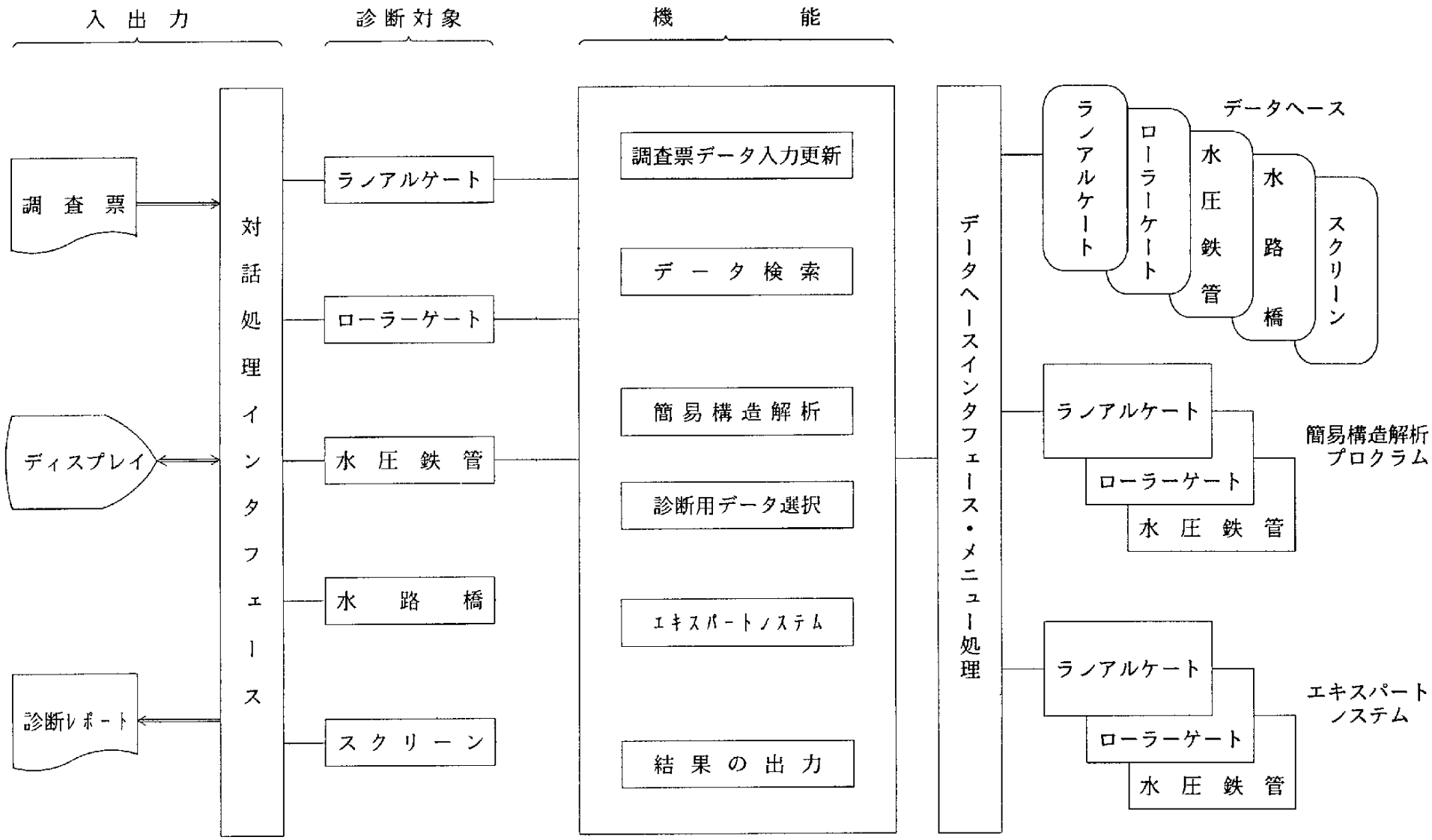


図44－1　HSDSの機能構成

システムの開発の経緯は表4 4-1 のようにまとめられる。ＤＧＤＡでは、第２版までが探査型プロトタイプの性格が強く、問題解決手法の妥当性の検討に当てられた。第３版は、実験型プロトタイプであり、これによって図4 4-1 の機能をほぼ全面的に実現し、システムの実験評価を行った。大型コンピュータ上に再開発した第４版からが、発展型プロトタイプである。一方、ＰＳＤＡは、ＤＧＤＡの経験をもとに、最初から発展型プロトタイプとして開発され、２つの版が開発された。いずれも、知識ベースの構造・推論方式は、ＤＧＤＡとほぼ同しである。これらを統合したＨＳＤＳは、現場での評価を繰返しながら、現在、実運用化の試行にいたっている[中村 1990]。

表4 4-1 開発経緯のまとめ

1985年 4月	既存部分(DB,FORTRAN)開発
1985年 4月	エキスパートンステム化決定 (それ以前の計画では統計的手法を使う)
1985年 5月	一次案としてDGDAのルール作成 (IF-THEN + 基本的な計算式)
1985年 6月	DGDAデモ用プログラムをM200上に作成 (デモ終了後廃棄) (第１版)
1985年 9月	技術的検討
1985年 9月	ＫＥＥの導入
1985年10月	DB情報を元にDGDAダムゲートのフレーム化 ルールの追加・修正(第２版)
1985年11月	ルールの追加・修正(第３版) システムのテスト
1986年 3月	大型機上で実用システム開発(第４版)
1987年 3月	大型機上でPSDAンステム開発(第１版)
1988年 3月	大型機上でPSDA実用システム開発(第２版)
1989年 3月	大型機上でHSDSンステム統合化
1990年	実運用化試行中

4.4 1.3 問題のタイプとンステムの特性

本システムは、診断型エキスパートシステムであり、表4 4-2の特性をもつ。

表4.4-2 システムの特性一覧表

	診断型システム	関連性	実現性	コメント
定義	対象とするンステムに異常または故障が生じたとき、観測されたデータ及びシステムに関する知識を利用して、原因の箇所を同定することである。	○	○	典型的な診断問題である。ただし、寿命予測の面では、単純な線形外挿を行っている。
問題の特徴	1. 設計レベルの知識の利用	○	○	対象システムの構造知識
	2. 経験的知識の利用	○	○	ルールによる
	3. 経験的知識の不確実性	○	○	AHPによる
	4. 操業データの利用	○	△	アンケート情報の一部
	5. 計測のための時間とコスト	○	×	計測は非常にコストがかかる
	6. 知識の抽象化の必要性	△	×	フレーム／ルールで対応可
	7. 対話的診断の推論制御	×	×	
基本タスク	1. 異常現象の分類階層的表現	×	×	
	2. システム構造の階層的表現	○	○	フレーム、AHPの階層
	3. 解釈問題のタスクを内包	○	○	
	4. 計測点の選択的決定	×	×	
	5. 異常原因の同定	○	○	
	6. 浅いモデルによる効率性	○	○	ルール
	7. 深いモデルによる完全性	○	○	数値計算による
問題解決機能	1. 事象駆動型推論	○	○	Ver 4はすへて前向き推論
	2. 目標駆動型推論	×	×	
	3. 不確実性推論	○	○	AHPとﾙｰﾙ/ﾌﾚｰﾑの組合せ
	4. 仮説推論	×	×	
	5. 協調型推論	×	×	
	6. 効率性と完全性の調和	○	○	ｼｽﾃﾑ自身診断の効率向上を目指している
	7. 費用／効果分析	×	×	

4.4.1.4 客観的な評価の可能性について

ＨＳＤＳは予防保全を目的とする診断システムである。そのために、ンステムが下した結論の正しさを実地において確認するという形ての評価は原理的に不可能である。また、従来のダム構造物に対する事故例もごくわずかしかないため、システムの診断結果を統計的に処理し、その信頼性を判定することもできない。

したがって、システムの客観的な評価は、知識ベース中の知識が利用者にとって妥当なものであるか、また、利用者インタフェースが適切なものであるか、統合性、発展性の意味で成熟したシステムとなっているかでなされる。これには、ソフトウェア自身の開発とは離れた部分でのシステムの権威付けが必要であり、コンセンサスを得るまでに非常に時間がかかるものとなっている。

4 4 2 ガイトラインの適用結果

（１）知識ベースの妥当性評価

本システムの知識ベースは、部分的に存在する診断評価基準を、専門家の経験則に基づいて整理したものである。知識ベースは比較的小さく、また、推論ステップも短いので、全数チェックに近い形でシステムのテストを実施することができた。知識そのものが妥当であることを利用担当者に確認させるためには、上位レベルの診断知識をドキュメント形式で提示することとした。下位レベルの知識を評価するためには、上で述べた理由からＡＨＰを利用した。

知識ベースの正当性を評価するために、ＤＧＤＡ、ＰＳＤＡともに実用バージョン開発の過程で特に以下の項目に準じた評価を実施した。

(C01)知識源の確認：知識源として、規準・実測データ・専門家の経験的知識の明確化を行ない、これらの間に整合性が保たれるように留意して知識ベースを作成した。

(C02)知識の型付け：採用したツール(Eshell)の特性に合せて分類を行なった。

(C04)専門家によるレビュー：ツールの提供する表現形式では、可読性に欠けるので、それをドキュメント化した形式で確認した。

(C07)専門家間での合意：知識の重要度を評価するために、複数の専門家によるＡＨＰの適用を試みるとともに、委員会形式て利用者レベルの専門家の意見の合意を得た。

知識ベースの完備性評価については、知識ベースの規模が大きくはない理由からテストデータを用いたＶ＆Ｖ((C11),(C12)) を行なった。

知識ベースの一貫性評価については、対象モデルの変化が少ないことから、本格的なＶ＆Ｖは必要ないと考えられた。ただし、上位レベルの診断知識は、下位レベルと分離することで、今後の保守作業の容易化を図った。

経験的知識を正当化する方法・確認については、上に述べた知識ベースの正当性評価と同様の手続きにより、(C01) 第３者専門家の意見に基づいた確認を行なうとともに、(C05)実績データとの照合によった。

知識表現の選択と適切さの確認作業については、ＤＧＤＡの第１，２，３バージョンの開発・評価過程において十分な分析を実施した。そのため、ＰＳＤＡ開発にあたっては、この問題は考慮する必要は生しなかった。

(C01)各知識表現の得失：第２バージョン開発において、ルール・フレーム形式を用いることに決定した。

(C02)知識表現の選択条件，(C05) 知識表現と推論機構：実用バージョンにおいて、ＨＳＤＳトータルの性能が十分発揮できるような表現を選択した。

知識獲得方法の選択と確認については、知識源の妥当性（(C01)専門家の選定、(C02)資料の適切さ、(C03)専門領域知識の収集）について十分な注意をはらった。また、事例分析・調査を通じて、ＤＧＤＡ第３バージョンの完成までに、典型例について、(C12)テストケースの収集を行なった。

（２）問題解決・推論方式の妥当性評価

ＨＳＤＳは、対象知識の表現にフレームを用い、推論には前向きのルールを利用している。さらに、不確実な情報処理を容易化するためにＡＨＰを利用した。本ンステムで、ＡＨＰがうまく適用てきたポイントは次の２点にある。ます、知識処理技術の面からは、フレーム概念によって、知識の階層的整理が容易に行えたこと、ルール概念によって経験的な意思決定知識を記述てきたことが上げられる。一方、ＡＨＰの面からは、評価を手軽に行えたこと、知識の整合性を検証することができたことが上げられる。

推論処理の妥当性評価については、推論処理方法があまり複雑てないため、チェックリストに示された項目に関する本格的な評価は行なっていない。推論時間の妥当性、推論手法の選択評価については、ＨＳＤＳては問題とはならなかった。

不確実情報の処理については、(C03)不確実性を用いないアプローチの検討、(C04)不確実性の意味の明確化に留意して、前述したように、ＡＨＰを用いる方法を開発し、これをＤＧＤＡ第３バージョンから適用した。

学習機能はＨＳＤＳでは採用していない。

（３）結果の妥当性評価

ＤＧＤＡの実験評価は、第３版システムを使って最初に実施された。それには、すでに修理・交換ずみのデータを含む、専門家による調査が終了したダムゲート25ヵ所の情報が使用された。この結果、システムの診断結果は、従来の専門家のものよりもやや厳しくなることが確認された。以後は、新しい測定結果の入手、ならびにシステム試用の要請にしたがって、ＨＳＤＳ全体を稼働させ、知識ベースを少しづつ改良することで、より妥当な結果が出るようにンステム評価を継続している。

正解の評価については、(C01)正解の定義、(C03)例外データの準備、(C06)正解の吟味、(C07)解に対する合意形成、に十分な時間をかけた。これには、従来型ソフトウェアにおけるのと同様なテスト手法を採用した。結果を定量的に分析する必要性はなかった。

解の網羅性・最適性については、診断問題ということもあり、ＨＳＤＳては特にチェックを行なわなかった。解の評価方法については上に述へたとおりてある。

（４）要求仕様に対する妥当性評価

本システムの開発は、専門家が中心となっていたため、利用者からの要求仕様を明確化する作業は、発展型プロトタイピングの最終段階て行われた。そのため、とくに、要求仕様に対する妥当性の評価は行っていない。

（５）知識ベースの保守性評価

ＨＳＤＳの知識ベースは、鋼構造物の構造、ならびに、構造解析の視点から自然に定まる部分と、診断意思決定に関連する不明瞭な部分とからなる。そこて、知識ベースの保守を容易化するために、意思決定にかかわる部分のルールを利用者にわかりやすい形でトキュメント化している。また、ンステムの開発は第４版システムからは、一貫して同一担当者があたっており、通常のソフトウェア開発で作られるような開発トキュメントは整備されている。ただし、これらについて

は、チェックリストに示されたような厳密なチェックは行なっていない。

（６）ヒューマンインタフェースの妥当性評価

ヒューマンインタフェースの評価は、利用担当者が第４版システムを試用する過程で行われた。第４版システムは、大型コンピュータ上でＴＳＳを使って実現されているため、インタフェースの機能は十分ではない。特に、レスポンス時間、図形出力、レポート出力などの点で不満が見られた。

利用時のインタフェースについては、結果の表示機能・操作性、カイダンス機能、エラー処理について改善を試みているが、チェックリストに対応する項目は少ない。

保守時のインタフェースは特に考慮していない。

（７）システムの総合評価

システムの総合評価は、知識ベースの妥当性を専門家・管理者間で検討する過程と、システムを利用担当者が試用する過程とで行われた。その結果、システムのもつ知識に対しては、ほぼコンセンサスが得られており、また、システム機能としては測定データと構造解析結果とを比較対照できる点が評価されている。

特に、チェックリストに照し合せて評価を下す必要性はない。

《参考文献》

[中村 1986] 中村秀治，松浦真一，寺野隆雄，篠原靖志：水力鋼構造物の寿命予測エキスパート・システムとその適用，土木学会論文集，No 374/I-6 (1986)，pp 513-521

[中村 1987] 中村秀治，寺野隆雄：土木構造物エキスパートシステム，オーム社，1987.

[中村 1990] 中村秀治，山本広祐（他）：水力鋼構造物・評価診断システムの試験的利用について，電力土木，1990.

4.5 ガイトラインの評価

本章の4つのエキスパートシステムの評価例は、本ガイドラインの実際のエキスパートシステムへの適用を試みたものであり、本ガイトライン自体の評価の目的が強い。以下に、その評価のポイントを、有用であった点と、今後の課題として残された問題とに分けてまとめる。

(1) 有用であった点

①チェックすべき項目が網羅されている。
②従来は気づかなかった評価視点が明確になった。
③うまくいっていない点検項目につき検討することにより、そのエキスパートシステムに内在する問題の本質が明確になる。
④評価の役割分担の指針が与えられているため、それぞれの立場て評価てき、また結果的に評価作業の効率が良くなる。
⑤評価作業の総量を見積ることができ、計画的な評価作業が行える。
⑥プロトタイプでのチェックポイントが充実しており、特にプロトタイプ版から実用版に発展させる場合に有効である。
⑦ライフサイクルに沿って分類がされており、それぞれの段階での評価の視点が明確になる。
⑧エキスパートシステムの設計時のチェックリストとしても利用でき、またシステム改善時の指針が得られる。

(2) 今後の課題

①評価を行うには、評価項目、評価方法、評価基準の3つの要素が必要てある。本ガイトラインでは、評価基準が設定されていないため、個々の評価項目はチェックしても、総合的な評価の判断が行いにくい。
②点検項目は網羅されているが、反面、項目が多すぎ、優先度や重要性の区別が難しい。
③測定項目が一部に偏在している。例えば、知識ベースの妥当性評価には何等かの測定項目が必要ではないか。
④エキスパートシステムの型別などの配慮は行われているが、欲を言えば、より対象問題別のガイトラインが存在することが望ましい。今後、このガイトラインを応用分野固有のガイドラインの作成に役立てることが出来そうてある。
⑤点検項目をシステマテックに数量化するなど、効率的かつ総合的な評価を可能とする手法の提供が望ましい。
⑥本ガイドラインをツール化し、コンピュータによる支援（例えばエキスパートシステム評価用エキスパートンステム）が行われることが望ましい。
⑦知識獲得や知識レベルアナリシスに関わる評価の視点が希薄てある。

5. 結　　論

5. 結 論

本報告書では、エキスパートシステム評価ガイドライン研究委員会の最終報告をとりまとめ、大規模なエキスパートシステム開発・運用のライフサイクルを通じて適用可能な、エキスパートシステム評価ガイドラインを提案した。本ガイドラインは、３４０項目におよぶチェックリストを中心としたものである。これには、エキスパートシステム開発・運用時に評価すべき重要な項目がすべて網羅されている。

本報告書では、このガイドラインの基礎となる諸概念について、知識工学的視点・ソフトウェア工学的視点から解説を行うとともに、チェックリストの利用方法について説明した。さらに、具体的なエキスパートシステム構築例に基づいたチェックリスト自身の評価を実施した。

本調査研究の主要な成果は次の５点にまとめられる。

（１）実際に運用されるエキスパートシステムの性質や、開発のライフサイクルまでを考慮に入れた、評価ガイドラインを提案したこと。

（２）エキスパートシステム開発・運用時に評価すべき重要な項目をすべて網羅した評価のためのチェックリストを作成し、その利用方法を取りまとめたこと。

（３）チェックリストを、点検項目・測定項目に分類してそれぞれ詳細な説明を加えるとともに、それに、ライフサイクル・評価対象・評価分類・テスト手法・システム利用形態・対象とする問題のタイプ・開発グループ・知識獲得活動・ドキュメントの９つの視点から分析を加え、使いやすい体系に整理したこと。

（４）エキスパートシステム評価に関する代表的な理論・手法について広く調査研究を行い、重要な文献を紹介するとともにリストにまとめあげたこと。

（５）具体的なエキスパートンステム開発事例に本ガイドラインを適用して、その有用性の評価を行ったこと。

エキスパートシステムの評価技術・手法について、本報告で論したような本格的な調査研究は、我国でははじめてのものと信ずる。また、このような研究は、世界的にみてもきわめて少ない。本報告書ならびに本報告書で引用した諸文献を参照することによって、各組織において、評価手法を重視したエキスパートンステムの開発方法論を整備することが可能となるとともに、評価手法そのものに関する最新の情報を得ることが可能となるであろう。

しかし、エキスパートンステムの評価手法、あるいはＶ＆Ｖ手法の研究は、まだ始まったばかりであり、技術的にも未熟な点が多い [O'Keefe 1989], [Culbert 1990]。本報告で示した評価のガイトラインも、網羅的かつ大規模なものだけに、実際の適用においては、改善の余地は多い。すなわち、本報告書の内容は、あくまで一般的な意味でのガイトラインてょり、本ガイトラインの利用にあたっては、各自が、自らの問題に応して各項目の内容を吟味し、項目を取捨選択して自分の問題に合せていく必要がある。

人工知能技術の理論的研究の進歩は著しく速い。一般のソフトウェアの評価技術も確立していない現状において、現時点で、本報告書のような評価ガイトラインを作成するのは時期早尚であるとの意見もあろう。しかし、我々は、この２年

半の間に、我国第一線の技術者・研究者が集まり、エキスパートシステム評価に関する大規模な調査研究を実施できたこと、また、本報告書に述べるようなガイドラインを作成・提案できたことは、きわめて意義が深いと考えている。本報告書の成果が、今後の知識システム技術の実践・研究の場で利用されることを願ってやまない。

本プロジェクトは、この報告書をもって終了するが、残された課題として、本ガイドラインをより実践的なものにするためには、エキスパートシステムのタイプについて、いっそうの分析が必要である。その際には、Generic Tasks にしたがった分類[Chandrasekaran 1986] も有用と思われる。さらに、今後、エキスパートシステム評価手法を確立していくためには、知識獲得支援ツールと評価手法とを関連づけた研究開発を実施していくこと、ならびに、ここで提案した評価ガイドラインをツール化していくことも重要である。

《参考文献》

[O'Keefe 1989] O'Keefe,R.M., O'Leary, D E The Verification and Validation of Expert Systems, 11th IJCAI Tutorial, MP4, 1989

[Chandrasekaran 1986] Chandrasekaran, B Generic Tasks in Knowledge-Based Reasoning: High-Level Building Blocks for Expert System Design. IEEE Expert, Vol 1, No 3, pp 23-30, 1986

[Culbert 1990] Culbert, C (ed) Special Issue Verification and Validation of Knowledge-Based Systems Expert Systems with Applications, Vol. 1, No 3, 1990.

付　　録

付録A. 海外調査報告

本委員会の下記のメンバーは、１９９０年９月にカナダ、米国に出張し、エキスパートシステム評価ガイドラインに対する意見交換、およびＩＥＥＥが主催したＭＥＳ’９０（Managing Expert System Programs and Projects Conference）で、特に評価に関する調査を行ったのでその報告を行う。

1. 調査概要

(1) 調査期間： 1990年9月2日（日）～9月16日（日） （15 日間）

(2) 調査員 ： 寺野 隆雄（筑波大学大学院経営システム科学専攻 講師）
石垣 良信（日本アイ・ビー・エム㈱ システムセンター先進テクノロジー 担当マネージャ）
宇佐見仁英（富士通㈱ システム本部第三システム統括部科学システム部第二システム課課長）
渡辺 正信（日本電気㈱ Ｃ＆Ｃシステム研究所システム基礎研究部研究課長）
金剛寺英雄（財団法人 日本情報処理開発協会 開発研究室 専任調査役）

(3) 調査対象期間
①９月４日（火） ブリテッシュコロンビア大学のBenbasat教授を訪問
②９月５日（水） シムフレックステクノレッジ社のHayes-Roth博士を訪問
③９月６日（木） 南カルフォルニア大学のO'Leary助教授を訪問
④９月７日（金） ｽﾄﾗｸﾁｬｰﾄﾞｼｽﾃﾑ＆ｿﾌﾄｳｪｱ社のGreen副社長を訪問
⑤９月10日（月）～９月12日（水） MES'90国際会議に参加
(Managing Expert System Programs and Projects Conference)
⑥９月14日（金） ジョージワシントン大学のLiebowitz助教授を訪問

(4) 調査概括

政府機関レベルでエキスパートシステム構築のための評価ガイドラインを作る試みは、非常に高く評価された。また、ライフサイクル毎にきめ細かいチェック項目を設定する方法も高い評価が得られた。準備不足のため（チェックリストの英訳など）具体的な内容まで突っ込んだ議論が出来なかったことは残念ではあるが、何れの機関も今後の継続的なコンタクトを望んでおり、海外調査の目的は果たせたと思う。

Hayes-Roth博士などのコンサルタントの立場と、O'Leary 助教授のような大学研究者の立場の違いが明確になった。AAAIなどで提言されているＶ＆Ｖの方法は、現実の場にはほとんど適用されていない。

ＩＥＥＥの国際会議は、技術的な議論が少なかったが、評価を研究している研究者や軍関係の人達と数多く接触でき、有意義であった。

基本的には、我々の想定しているアプローチで総合的なガイトラインが作成できることがわかった。

今回の調査で、ＶＶ＆Ｔと知識獲得との関連性が深いことを認識した。
従来エキスパートシステムの開発は、図1-1のライフサイクルを想定していた。

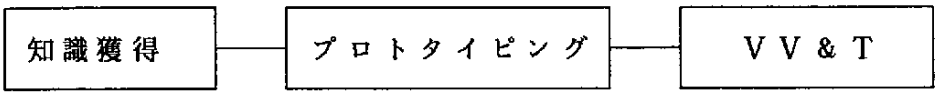


図1-1　従来のエキスパートシステムのライフサイクルの考え方

この様なアプローチは一般的ではあるが、ライフサイクルの各工程に対して、知識獲得をし、その結果に対しＶＶ＆Ｔを行う、図1-2のライフサイクルのモデルのほうが望ましいという印象を受けた。つまり、各工程で知識の獲得とその工程で作業した内容についてＶＶ＆Ｔを行う。

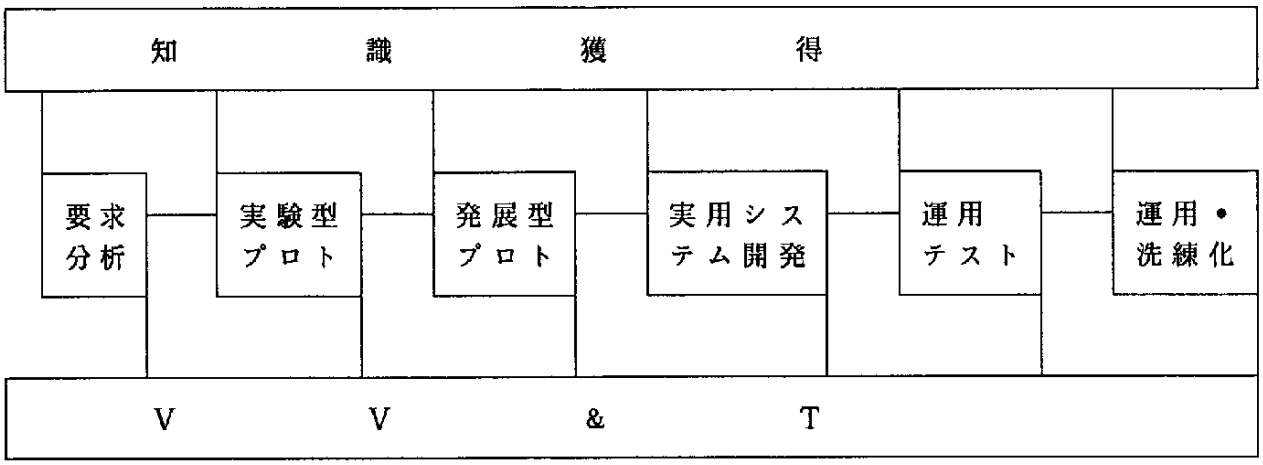


図1-2　エキスパートシステムのライフサイクル（新アイデア）

2. 本ガイドラインに対する意見交換

2.1 ブリテッシュコロンビア大学

①訪問日時：	9月4日（火）14 30-17:00
②面会者 ：	Izak Benbasat教授 Management Information Systems Jasbir Singh Dhaliwal Management Information Systems
③場所 ：	Faculty of Commerce and Business Administration University of British 2053 Main Mall, Vancouver, B C , Canada V6T1Y8
④主要論文：	I Benbasat, J S. Dhaliwal' A Framework for the Validation of Knowledge Acquisition, Knowledge Acquisition, Vol 1, No.2, pp.215-233, (June 1989)

我々のＶ＆Ｖプロジェクト内容について説明し、この内容についてコメントをして頂く形式で討論を進めた。なおBenbasat教授は、知識獲得が専門であるが、意思決定システム、ＳＩＳなどＡＩ応用に対しても興味が深い。以下は、討議の要点である。

- ライフサイクル毎のチェックリストは大変興味深い。
- 具体的なチェックリストになっていないのであまりコメントできないが、チェックリストは抽象的になりがちなので注意を要する。
- 知識獲得が主な研究でＶ＆Ｖはメインの研究ではないが、同一の問題に対し、別々に知識獲得を行い、その内容について比較検討をしたことがある。
- Ｖ＆Ｖと知識獲得は互いに関係が深いことが指摘された。
- Evaluationは状況に依存するが、Validationは技術的な観点から検討が可能であるという意見。
- VerificationよりもValidationの方が今後重要になると考える。
- 我々のＶ＆Ｖプロジェクトが設定したエキスパートシステムの定義はワイドレンジ過ぎるのではないか。
  専門家の代替システムを含めると、新しい状態（全く新しいシステム）では専門家がいないことになる。専門家がいることを前提とした支援システムとするべきではないかという意見。
- 評価で難しいのは、どのような品質解析手法(Quality Analysis)、評価手法を用いるかである。
- 要求記述(Requirement description)は、ユーザ毎、フェース毎に視点が異なる。例えば、ユーザは問題解決の入力と結果に対する要求記述をする。専門家は、知識に対する要求記述（機械に依存しない形て）をする。
- ヒューマンインタフェースでは、エントユーザにとっては説明機能が重要だが、専門家（またはＫＥ）にとっては分かりやすい知識の追跡機能が重要である。
- 最終ステージの決定のための第三者による評価がいるのではないか。
- Objective Evaluation of ESsを実践するためには、Knowledge Auditorが必要になるのではないか。そのスキルや教育方法はとうあるべきか。
- 問題解決法とタスクのタイプについては、非常に相互の依存性が高く、タスクの集合として問題解決法が決められる。

2.2 シムフレックステクノレッジ社

①訪問日時：	９月５日（水）14:00-17:00
②面会者　：	Frederick Hayes-Roth博士(Vice President) Richard S Brooks博士　(Manager)
③場所　：	Cimflex Teknowledge Corporation 18110 Embarcadero Road, Palo Alto, CA 94303, U S A
④主要論文：	F Hayes-Roth: Towards Benchmarks for Knowledge Systems and Their Implications for Data Engineering, IEEE TRANSACTIN ON KNOWLEDGE AND DATA ENGINEERING, Vol 1, No.1, March 1989

我々のＶ＆Ｖプロジェクト内容について説明し、この内容についてコメントをして頂く形式で討論を進めた。以下は要点である。

- いままで、顧客から依頼されて作成したシステムについて、９０％は従来型の技術、残り１０％はＡＩ技術を用いてシステム化されている。つまり、良構造の問題を選んで、ルールベースシステム開発を請け負うことによって、ＡＩ企業として活動している印象であった。
- IEEEの論文にDARPAが興味を持ち、ベンチマークに関するプロジェクトを現在計画中である。
- 本ガイドラインのV&Vのライフサイクルは、リサーチ型に近いと思われる。この形での開発はUnhealthyであるという指摘があった。
  - タイプとして、リサーチ型とコマーシャル型のエキスパートシステムがある。我々がコンサルタントしているのは、後者である。
  - ５年前は、米国でも貴方方が想定しているようなリサーチ型のライフサイクルであったが、現在ではＡＩ市場は成熟しているので、Ill-definedの世界も我々はコマーシャルベースで行っている。
  - 米国では、産業技術がピークを過ぎているために、エキスパートンステムの需要が強い。
- プロジェクトの予測誤差をコスト換算すると５－６％である。５年前は、１０％であった。
- 当社におけるエキスパートシステムのライフサイクルは、図２2-1の通りである。

study	initial system	expand	expand	

図2.2-1　ｼﾑﾌﾚｯｸｽﾃｸﾉﾚｯｼﾞ社におけるｴｷｽﾊﾟｰﾄｼｽﾃﾑのﾗｲﾌｻｲｸﾙ

- Initial systemの段階でフィールトテストまで行う。この段階でエンドユーザが介入し評価する。expandするときは、１回ごとに顧客と契約をし直す。
- インクリメンタルに開発するが、プロトタイプを作成してjudgeすることはやっていない。
- プロトタイピングのアプローチは、リスキーである。プロトタイプを顧客にみせると過大な期待をすることになる。
- 調査／計画の段階で、顧客のマネージャが検討に加わるへきてある。

2.3 南カルフォルニア大学

①訪問日時：	9月6日（木）9.00-14:00
②面会者　：	Daniel E O'Leary助教授（博士）
③場所　　：	Graduate School of Business University of Southern California Los Angeles, CA 90089-1421, U.S.A
④主要論文：	D.E O'Leary Validation of Expert Systems with Applications to Auditing and Accounting Expert Systems, DECISION SCIENCE, Vol.18, No.3, pp 468-486, 1987

エキスパートシステムのV&Vの概説と問題点を講義して頂くと共に、我々のプロジェクトに対しコメントをして頂いた。

2 3.1 概説と問題点

(1) ルールベースシステム（RBS）の検証技術

- 事象はどのくらいの頻度で発生するのか、他の事象との相関はどのくらいなのかを統計的手法を用いて分析する。
- そのルールの有効性を検証するには、そのルールが役に立たないことを実証すればよい。
- ドメインの専門家による検証
  例）飛行機のフラップの上下の問題では、専門家による事象発生の組合せを抽出する。
- 循環ルールの発見

(2) RBS以外に対する検証技術

- 即ち、フレーム、オブジェクト、セマンテックネット、非単調推論などのことである。
- フレームでは、スロット数、スロットの内容、フレーム間が循環、多重継承等を検証する。

(3) ライフサイクル

(3-1) 開発の主体が、①エンドユーザ、②コンサルタント、③従来の情報システム部門によりライフサイクルに対する考え方が異なる。
特に、コンサルタントが想定するライフサイクルは、初期にリリースしたバージョンを契約により、次々とバージョンアップしていくやり方を取る。バージョンアップするとき、どのくらい効果が上がるのか、予算がどのくらいなのかがポイントである。

(3-2) ライフサイクルは次の項目に影響する。

- 開発者の最終目標
- 組織上の制約
- システムは誰が使うのか
- 計算機の知識

• 応用分野の成熟度合
• システムはどのくらい理解されているか

(3-3) 研究開発するときは、プロトタイピングアプローチを採用し、従来のソフトウェア開発では要求分析のアプローチから始まる。この２つのアプローチをマージすることが私の立場である（同意見）。

(4) Ｖ＆Ｖをこえて
• アセスメント、即ちコスト分析、利益分析、セキュリティ等の問題がある。
つまり、よいシステムなのか、よりよくしてくれるンステムなのかの問題である。

(5) 知識ベース上の議論
• 推論エンジン
• エラーの発見の仕方
• 推論は正しい？
• 決定的な問題か？

(6) Ｖ＆Ｖと知識獲得との統合化
• 複数の専門家からの知識獲得の問題
• ワークショップに出したペーパー（１件）
「Deciding which experts to use . in the case of multiple experts」
この中で、４人の専門家がいて(a,b,c,d)、４人とも異なる世界ビューを持つ。但し、(a),(b),(c)は似たビューを持つ。この場合、(a,b,c)or(d)のどちらを採用するのか。また誰が決めるのか。
• 専門家が学んだ学校（学派、流派）の違いをどうシステムに組み込むのか。
• 自動化を狙うエキスパートンステムは、早期の段階から検証すべきである。
• 信頼性の問題（信憑性）
もともと専門家は、Ｘを思っていてもＸに近いＸ’の程度しか言わない。
Ｘ’／Ｘ＝知識獲得の信頼性

(7) 重み付けの問題
• 重みに関する表現は、ＫＥにとってもよい表現とは言えないことてある。
• ５つのシステムの構築を経験したが、各々は重みに関して過ちがあった。
• 確率論、重みの意味付けを踏まえて重み付けすることはなかなか難しい。
• 類似の問題として、天気予報官のだす降水確率がある。
• ベイズの確率の適用例：ＡＬ／Ｘ、Prospectorなど

2 3.2 O'Learyに対する質問

(1) エキスパートンステムと従来ンステム
• Hayes-Roth博士のようにコンサルタントをしている場合、９０％は従来システムを対象にしたシステムである。

(2) 分類基準（ライフサイクル、開発チーム、問題のタイプ）

- 分類基準は、当然考慮すべきであり、色々な観点から検討しておくことが重要である。

(3) チェックリストは有効か?

- チェックリストは役に立つが、誰が使うのかを明記しておくこと。

(4) 定量的な評価項目は可能か?

- 役に立つものは知らない。

(5) 評価をサポートするツールとは?

- 知識獲得用のツールが評価をサポートするツールに近いため、それに着目した方がよい。
- ルールの構造化
- 重みの問題

(6) 本ガイトラインの作成は、Too Ambitiousであると思う。1991年の7月にV&V on ES Workshopを開催するので、ぜひ参加して頂きたいという要請があった。(AAAI'91, Innovative Application on AIのときらしい)

2.4 ストラクチャードシステム＆ソフトウェア社

①訪問日時：	９月７日（金）9:30-13.00
②面会者　：	J. Ellis McSparran社長 Christopher Green副社長 Marlene M　Eckertｼｽﾃﾑｴﾝｼﾞﾆｱ
③場所　　：	Structured Systems & Software, Inc. 23141 Plaza Pointe Dr , Laguna Hills, CA 92653-1425
④主要論文：	C.J.R　Green, M M. Keyes' Verification and Validation of Expert Systems, WESTEX-87, pp 38-43, (June 1987)

2.4.1 討論の要約

最初に、Green氏により当社におけるＶ＆Ｖについての考え方及び、我々のプロジェクトについてのコメントを頂いた。その後、日本におけるエキスパートシステムの開発利用状況について説明を行った。以下は、Ｖ＆Ｖに関する要点である。

- Ｖ＆Ｖは、非常に専門的なビジネスである。
- ＮＡＳＡが開発／保守したＣＬＩＳＰ（プロダクション型シェル）を用いて、エキスパートシステム開発の教育とValidationを実施している。
- 評価は、カウント可能な機能単位に分割して実施している。すなわち、実行コード（手続き）、データ（オブジェクトを含む）、ルール等に分割する。
- ルールの複雑性をクラス分けして設定している。
- チェックリストによるアプローチはよい方法(Reasonable,Bestway)である。
- また、チェックリストについては従来のソフトウェアではあるが既に商用化され、ライセンス化されている。従ってこれらを使用するときは、要注意である。
- チェックリストは、経験した事柄を踏まえて改訂することが重要である。

2.4.2 Ｖ＆Ｖコンサルタントのコンタクト先

(1) RADC Program Manager
　　Capt. Arlan Morse
　　RADC/DCLD
　　Griffiss AFB, N.Y. 13441
　　morsea@lonex radc.af.mil

(2) Lincoln Laboratory
　　Dr　Harold M. Heggestad
　　M.I.T. Lincoln Laboratory
　　244 Wood Street, Mail Stop M200
　　Lexington, Mass　02173
　　hal@xn.ll mit.edu

(3) 3S(Structured Systems & Software, Inc)
　　Ms. Marlene Eckert　　(Program Manager)
　　Mr. Christopher Green (Chief Engineer)
　　sss@lonex.radc af.mil

3.　ＭＥＳ’９０国際会議（９月10日（月）～12日（水））
Managing Expert System Programs and Projects Conference

【１】　国際会議の状況

メリーランド州BethesdaのホテルHoliday Innで、MES'90の国際会議が行われた。参加人数は、約１６０人であり、軍関係者（制服組）、ＮＡＳＡ等の公的機関の参加、大学関係者はビジネススクールの先生方が中心に参加していた。

この会議は、マネジメントの視点からの発表が多い会議であり、ＡＩ研究者の参加は少ない印象を受けた。そのためなのか、エキスパートシステムのＶ＆Ｖが、マネジメントサイドの議論に終止してしまう傾向があった。また海外からは、韓国、イタリア、ドイツ、イギリスが参加していた。また日本とアメリカとのマネジメントの仕方についての差も感じた。

エキスパートシステムの評価に関しては、「追跡可能性(tracesability)」の評価が重要であることが強調されていた。

【２】　チュートリアル２：エキスパートシステムの検証と妥当性検査

講演者：　R Glenn Wright
　　Graphic Media Arts　Inc.
資料名：　R G　Wright' Expert System Verification and Validation,
　　MES'90, Tutorial #2, September 10, 1990

（議論のポイント）
- ＤＯＤ－２１６７Ａなどののフルスケールドキュメントとミニマムトキュメントでは、どちらを採用した方が有効であるか。
- エキスパートシステムを開発する際、Ｖ＆Ｖは誰が行うのか。ＤＥ（専門家）では分からないのではないか。

（講演内容）

1.　検証(Verification)と妥当性検査(Validation)（Ｖ＆Ｖ）

検証とは、エキスパートシステムの開発工程において、正しい製品を構築することを検査することである。妥当性検査とは、エキスパートシステム製品そのものに対して、正しい製品であるのかを検査することである。

テスト(Testing)とは、プログラムの能力を評価すること、問題領域を設定した範囲内で専門家の優れた知識を確立して、改訂することである[Hayes-Roth83]。

評価(Evaluation)とは、特定の問題領域に関して、実世界の問題を解法するため、エキスパートシステムの能力を検査するプロセスである[O'Leary 90]。
評価とは、ソフトウェアの正しさや有効性を測る尺度である[Liebowitz 86]。

保証(Certification)とは、システムが適切に総合システムと相互作用をし、総合システム内で指定した機能の実行を確実に行う行為である。

2. エキスパートシステムのV&V

2.1 従来システムの開発のアプローチ

表2-1 従来システムの開発のアプローチ

[Deutsch 81]	[Metzger 81]	[Ramamoorthy 84]	[Dunn 84]
詳細分析	定義	要求の理解	システム仕様
アロケート	設計		ンステム設計
定義	プログラミング	要求仕様の記述	ｿﾌﾄｳｴｱへの要求
設計	システムテスト	設計	ﾄｯﾌﾟﾚﾍﾞﾙ設計
構築	受け入れ検査	インプリメント	詳細設計／コード
テスト	導入	テスト	単体テスト
統合化			統合化
運用	運用		テスト
メンテナンス		メンテナンス	ﾊｰﾄﾞ/ｿﾌﾄの統合化
		発展	システムの品質

2.2 エキスパートシステムの開発のアプローチ

表2-2 エキスパートシステムの開発のアプローチ

[Buchanan 83]	[Bobrow 86]	[Keller 87]	[Martin 87]
定義	定義	調査	要求分析
概念化	概念化	構造化分析	モデルの定義
		知識獲得	
		ﾊｰﾄﾞｳｴｱ分析	
定式化		設計	ｱｰｷﾃｷﾁｬの設計
		知識獲得	
		知識ベース設計	
ｲﾝﾌﾟﾘﾒﾝﾄ	ﾌﾟﾛﾄﾀｲﾋﾟﾝｸﾞ	ｲﾝﾌﾟﾘﾒﾝﾄ	ｲﾝﾌﾟﾘﾒﾝﾄとﾃｽﾄ
	ﾕｰｻﾞｲﾝﾀﾌｪｰｽ		
テスト	テストと再定義	許容テスト	
		システム統合化	評価
	知識ベースの更新		

2.3 ソフトウェア開発の各種プロセスモデル

ソフトウェア開発の各種プロセスモデルには、

- ウォーターフォールモデル
  - 同定
  - 概念化
  - プロトタイピング
  - ユーザインタフェース

テストと再定義
知識ベースのメンテナンス
• ラピッドプロトタイピング
要求仕様の記述とインプリメントがオーバーラップする。
• スパイラル型の開発
• 多層開発(Multiple Build)
等があり、これらを組合せる場合がある。

2 4 V & V

V & Vは、ソフトウェアのライフサイクルの全てのフェーズに適用する。
• 概念化設計
• 開発とテスト
• 運用
• メンテナンス
このためには、タスクの管理を技術的に行わねばならない。
• デザインレビュー
• 要求のトキュメント化
• 組織上の責任

エキスパートシステムは、不確実な推論、悪定義の領域を扱うなどのため、V & Vを行うのが難しい。

2.5 エキスパートシステムのV & Vを支援するトキュメントについて

エキスパートシステムを開発する際、トキュメントが全然ない場合、ミニマムドキュメントがある場合、DODの標準であるフルスケールのトキュメントがある場合など各種のレベルがある。正式なV & Vを行うには、ユーザが規定した要求仕様が必要である。

2 5.1 ミニマムドキュメントのアプローチ

危機的な状況でエキスパートシステムを使用することがなければ、ミニマムドキュメントで充分である。ミニマムトキュメントとは、要求仕様と知識仕様のことである。
要求仕様：エキスパートシステムが満たすへき要求の集合
知識仕様：エキスパートシステムの要求から得た知識の集合

(1) 要求仕様
解くべき問題の定義と成功基準の定義が述べてある。初期の段階に作成され、要求に応じて更新される。要求仕様の内容は以下の通り。
• エキスパートシステムの目標
• 開発の方法
• システムへの要求
• 専門的知識の源
• 期待される結果

これらは、下記に示す誰からの要求であるのかを明確にしておく。

- 顧客(Customer)
- プロジェクト管理
- 領域の専門家
- 開発者
- ユーザ

(2) 知識仕様

知識獲得の結果、得た知識を記述する。知識源を追跡することが、本トキュメントの初期の目的である。知識仕様は、知識獲得の行為により充実する。
知識仕様の内容は以下の通り。

- 知識源
- 知識の所在（生データ、ノート等）
- 知識のタイプ
- エキスパートシステムの使い方（如何に、何処で、何故）
- 他の知識との関係

なお、知識仕様はドメインの知識、知識獲得法に依存する。

2.5.2 フルスケールドキュメントのアプローチ

ＤＯＤのドキュメント標準(DOD-STD-2167A)に沿うアプローチである。この標準は、ウォータフォールモデル、ラピッドプロトタイピング、スパイラル型にも合う。このうち、下記の５つのドキュメントがエキスパートシステムに適用可能なドキュメントである。

- システム／セグメント仕様　（ＳＳＳ）
- システム／セグメント設計ドキュメント（ＳＳＤＤ）
- ソフトウェア開発計画　（ＳＤＰ）
- ソフトウェア要求仕様　（ＳＲＳ）
- インタフェース要求仕様　（ＩＲＳ）

2.6 エキスパートシステムの検証技術

検証(Verification)とは、エキスパートシステムの開発工程において、正しい製品を構築することを検査することである。

[Prerau 1990] 知識獲得の工程は自動的にソフトウェアのバグも含めて検証テストを実行することである。

[Lowry 1989] シビアなソフトウェアは、正式な仕様書に対して時々数学的に正しさを求めるために検証を行う。

検証する対象は、知識ベース、推論エンジン、ユーザインタフェース、データベースインタフェース、他のシステムとのインタフェースである。検証は、各開発フェーズで作成されたトキュメントと製品を比較して行う。

2 6.1 知識ベースの検証

専門家と協力して獲得した知識は，知識ベースに移植する際、繰り返し検査やテストが行われる。すなわち、知識に基づく推論をシュミレートしたり、知識に存在するエラーを検出したり、知識をインプリメントする際に生したエラーを検出したりする。

O'Learyは、妥当性検査を3段階の検証として位置づけている。すなわち、①Face Validityの保証、②サブンステムの正当性(Validity)、③入出力変換による比較である。①と②は、検証に近く、③は妥当性検査の範囲てある。

(1) プロダクンョンルールの検証

- 同し名前のルール名がないか。
- 構文的に誤ったルールはないか。
- 冗長のルールはないか。
- 孤立したルールはないか。
- 包摂ルールはないか。
- 競合ルールはないか。
- 循環ルールはないか。

これらは、市販の構築ツール上においても簡単な構文チェンクなどの機能はある。また、ルール、知識グループ、他の知識ベース要素との対応関係の視覚化等の機能もある。

(2) フレームベースの検証

- 非合法なスロットの値
- 同じ名前のフレーム名
- スロットの継承の競合
- 継承パスの循環

シビアなエキスパートシステム即ち、生命に係わるシステム、危機的な状況で使用するシステム、他のハイリスクのあるンステムに対しては、推論エンジンのV&Vをすへきである。従来のソフトウェアテスト技術を用いてブランクボックスとして推論エンジンのテストをすへきである。

他に、ユーザインタフェース、データベースインタフェース、他のンステムとのインタフェースの検証をすへきである。これらは、従来のソフトウェアの検証技術を用いることが出来る。

2.7 エキスパートシステムの妥当性検査技術

妥当性検査(Validation)とは、エキスパートシステム製品そのものに対して、正しい製品であるのかを検査することである。検証と共に、下記の項目ついても妥当性検査をすへきである。

- 知識ベース
- 推論エンジン
- ユーザインタフェース

• データベースインタフェース
• 他のシステムとのインタフェース

妥当性検査の方法は、①絶対的な妥当性、②専門家の能力と比較した妥当性、③フィールドテストによる妥当性がある。［Prerau 1990b］

(1) 絶対的な妥当性

• 数学的な問題
  正しさは証明による。
• 装置の診断問題
  システムによる自動化で、システムが正しく作動しているのかによる。
• 予想問題
  正しいと決定した過去のデータとに比較による。

(2) 専門家の能力と比較した妥当性

専門家が解に対してあまり正当性が明確でない分野では、決定が良かったのか、理由付けがあるのか、受け入れられるのかなと専門家間で合意しておくことが必要である。

• 医療診断問題
• 予想問題

評価方法としては、

• 運用している専門家の判定と比較
• テストケースの結果との比較
• プロジェクトの専門家との判定比較

がある。

専門家と比較するとき、①多数の専門家と比較するときの問題、②第3者専門家との比較の問題がある。①は、多数決、コンセンサス、専門家とエキスパートシステム間で同意または反対の判定基準の問題がある。②は、プロジェクトの専門家でも気がつかない微妙な問題を第3者専門家により、そのバイアスを検査することである。

(3) フィールドテストによる妥当性

実際にフィールドでプログラムを実行してみることである。

• 製品使用として初期からフィールドテスト
  エラーがあってもコスト的な損失が少なく危害が少ない場合に適用。
• 併行運用
  妥当性検査を完全にするまで併行運用を行う。

(4) 入出力の比較による妥当性

• 量的な比較
  数値データの観測が可能、制御されたデータの収集が可能な場合。
  －スペクトラム分析
  －統計的な手法
  －適合検査
• 質的な比較
  主観的な比較になる。コストやスケジュールがきつい場合、デモ用のプロト

タイプの開発によりふさわしいのかを判断する。
－チューリングテスト
－フェース妥当性検査
－予想妥当性検査
－フィールトテスト
－サブシステム妥当性検査

2.8 エキスパートシステムのV＆Vツール

(1) KEE(ｲﾝﾃﾘｺｰﾌﾞ社)

KEEには、バリュークラスとカーデナリティに対する検証メカニスムがある。

- バリュークラス
  スロットに対して許される値の集合のクラス。
- カーデナリティ
  スロットに含まれるがスロット値の数

(2) SimKit(ｲﾝﾃﾘｺｰﾌﾞ社)

SimKitは、分散イベント・シミュレーションモデルを構築する際に用いるツールであり、バリュークラスとカーデナリティに対する検証メカニスムかある。

(3) Validator [Kang 90]

ルール型のエキスパートシステムで、①プリプロセッサー、②構文解析、③構文エラーチェッカー、④デバッガー、⑤チェーンニンク関係のチェック、⑥知識ベースの完全性の機能がある。知識ベースの構文的、意味的チェックの機能がある。

- 予約語の非合法的な使用
- 決して発火しないルール
- 未使用な事実
- 未使用な質問
- 未使用な合法的な値
- 同じ質問の繰り返し
- 合法的な値を使用したルールたが誤ったインスタンスを作成したとき

(4) SEEK [Polatikis 85]

知識ベースの質を高めるため、次の２つのタイプの機能かある。

- ルールの一般化
  条件を取り除く。確信度を多くする。
- ルールの特殊化
  条件を追加する。確信度を少なくする。

(5) EVA [Cheng 90]

EVAは、アルゴリスムの変換、妥当性検査モジュールの２つから成り立っている。

①アルゴリズムの変換
エキスパートシステムの知識を妥当性検査モジュールに受け入れられるように変換する。
②妥当性検査モジュール
下記のチェックを行う。
- 論理
- 構造
- 意味
- 省略
- 振舞い
- ルールの計画(Proposal)

ＥＶＡは、下記のエラーに対処できる。
- 非一貫性
- 未踏着ルール
- 行き止まりルール
- 再帰呼び出し
- 変数の合法的な値
- データのタイプ
- 切り捨てるへき知識

(6) ＣＨＥＣＫ（ロッキート社）［Gupta 89］
知識の完全性、一貫性を自動的にチェックする検証ツールとしてロッキート社のエキスパートシステムシェルＬＥＳ用に開発された。ゴールトリブン、データドリブンの両方で稼働する。フレームベースのシステム対しても適用が可能である。

(7) Ｃａｓｅ　Ｖａｌｉｄａｔｏｒ　［Smith 90］
- テストケースの格納
- テストした状況の記録
- 知識ベースのカバレージ・アナライザー

2.9 エキスパートシステムのＶ＆Ｖの将来

(1) シビアな応用に
従来型のソフトウェアシステムは、航空管制、原子力、ミサイル発射などのシビアな応用システムを開発しているが、エキスパートシステムはこれらの分野への適用がなぜ難しいのか。経験的な知識は、公式に証明されない、全ての場合に対しては正しくないからである。このためには、何がキィとなるのか。要求仕様と知識の追跡可能性がキィとなる。追跡可能性を遂行するためには、証明、標準化、自動化が必要である。特に９０年代のテーマとして、Ｖ＆Ｖの自動化が課題である。
(2) 追跡可能性
知識仕様の定義は進化し続け、その使われ方も広範囲になる。自動化されたエキスパートシステム・ツールにより直接要求仕様にアクセスできる。

2.10 まとめ

ライフサイクルの初期の設計段階からV&Vが必要であること、V&Vをサポートするためにドキュメントの確立が必要であること、V&Vを実施するための技術、その技術を使用したケーススタディ、V&Vをサポートするツール、将来におけるV&Vについて述べた。

【3】 チュートリアル6：エキスパートシステムの見積の正当性評価

- 見積りの正当性評価は、開発ステージ毎に行う。
  Idea/Concept→First Prototype→Upgrade→Production System→Evaluation
- プロトタイピングは、両刃の剣であるが、ある種のリスクは回避できる。
- 従来型のコスト評価法との比較では、CIMとの比較がおもしろい。
  CIMは、個々のユーザニーズが複雑で難しい上に、それを統合化する必要がある。
- 従来型システムと比べてエキスパートシステムの正当性を評価するとして、複雑性の判断、不確実性の存在などがある。

【4】 キーノートアドレス

「エキスパートシステムのプログラムとプロジェクトの管理」
Dr. Ed Mahler, AI Program Manager, E.I Dupont de Nemours and Company

- エキスパートシステムはビジネスの問題で高度で生産的な解をもたらす。
- 不確実の状況でビジネスの意思決定をする問題に対処する。
- 知識ベースのライフサイクル

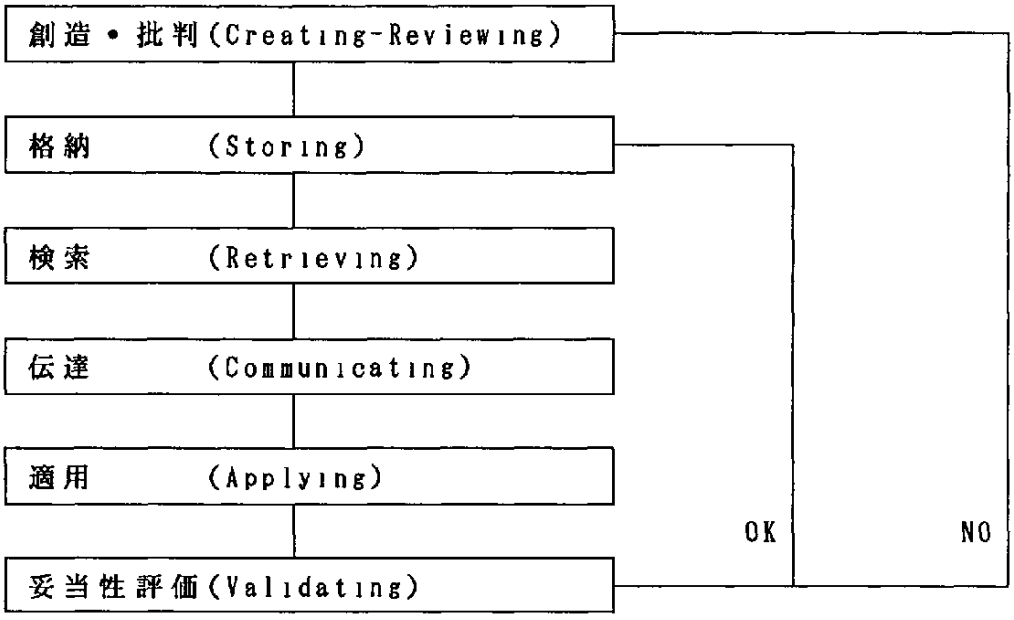


図25-1 知識ベースのライフサイクル

• ツールの選定

言語→シェル→アプリケーション→高度なアプリケーションとなるに従い、使用が容易となるためユーザは多くなる。その逆は、スピードが早くなると共に、フレキシビリティが増す。

【５】 PANEL3' Expert Systems Verification, Validation, and Testing

とりわけ興味深かったSAIC社のLance.Millerの発表内容及びそれに関する討議内容のポイントについて以下に述べる。

（討議内容のポイント）

• 高品質のエキスパートシステムの開発をするためにはどうすればよいのか。
• DOD-STD-2167Aの利点と欠点。
• ダイナミックテスト、リアルタイムテストは経費がかかる。
• コスト利益の観点でテストすべきである。
• 正解率を上げるというより、境界値テストをすべきてある。貴方がどのくらいの責任、資源、テスト用のドキュメントを持つのかてある。
• 不確定な情報などはドキュメント化すべきである。
• 結果に対してどのくらいニーズがあるのかを把握してから、小さな知識ベースでテストすべきである。
• テストは、実際の環境でテストすることが重要てある。
• データフローダイヤグラムなどのＣＡＳＥツールを利用してテストすべきてある。

（発表内容）

1　ソフトウェア開発用ドキュメントDOD-STD-2167Aの評価

1.1　利点

• 長年を経ている２１６７のドキュメントとなじみ深い。
• 明確に開発管理について述べてある。
• 追跡可能とするため機能分解に着目している。
• ｿﾌﾄｳｪｱｴﾝｼﾞﾆｱﾘﾝｸﾞで考慮すべきことを詳細に述べてある。
• システムの能力テストと評価に着目している。
• 構成管理を実施する。

１２　欠点

• かなり複雑で、時間を消費する。
• あまりにも多くのレポートを記述する。
• あまりにも多くのレビューを実施する。
• 安定した良定義問題向きであり、要求がすく変わるような問題には向かない。
• 手続き型言語向きである。

2　知識ベースシステムの管理（従来システムとの違い）

知識ベースシステムの管理において従来システムとの違いは以下の通りである。

①アプリケーション間で再利用するコードが従来に比較して多い。
（推論エンジン、シェル、外部インタフェース）

②従って初期の知識ベース要素の段階で管理をすべきである。
（より楽に開発ができ、トキュメント、テスト、修正が行える）

③知識ベースの要素は、より優れた開発技術の影響を受ける。
（高度な知識獲得技術、知識表現技術）

④知識ベースの要素は、より効果的なテスト技術の影響を受ける。
- 例外検出の自動化（ロッキート社で開発中）
- ルールの動的な構造テストに対するﾃｽﾄｹｰｽの生成（ｼﾞｪﾈﾘｯｸﾃｽﾄ法、SAIC）
- 新Critical Failure-Mode（SAICで開発中）
- 新知識ベース記述の自動生成（ﾙｰﾙとﾌﾚｰﾑの生成と静的なテスト、SAIC）
- 従来のテスト技法の集大成（SAICで開発中）
- 回帰テストも含めたテスト量の増大化

⑤知識ベースの要素はそれ自身機能分解の対象とはならない。
- 一般に、知識ベースの要素は多機能、多目的であるから。
- 特に、ルールベースの知識ベース要素は、モジュール化しずらい。
- しかし、フレームベースの知識ベース要素は、階層構造化される。

⑥多くの知識ベースシステムは、悪定義要求から始まる。
- 多くの局面があることは、開発当初は知らない。
- 他の局面はトラマチックに変化する。
- このため、よく定義され安定した要求仕様を苦心して作ることとなる。

⑦多くの知識ベースシステムは、ユーザやデータと対話する意思決定システムである。

⑧知識ベースの大半は、それ自身がドキュメントの役目を果たすことが出来る。
- ルールは、グラフ表現が出来る。

⑨知識ベースのトキュメント全体は、単純にすべきである。
- 詳細なトキュメントは、開発効果を減じる。

⑩知識ベースシステムにおいては、要求仕様や設計の表現はより簡単にてき、知識ベースのインプリメントと一貫性を保つことが出来る。
- フレームによる構造化、ルールによる条件分岐。

⑪知識ベースシステムにおいては、レビューすることが減少する。
- その理由は、知識ベースの要素の再利用、ﾗﾋﾟｯﾄﾌﾟﾛﾄﾀｲﾋﾟﾝｸﾞによる要求、開発、V＆V、設計、インプリメントの減少のため。
- 知識獲得、テスト方法が強力であるため。

⑫知識ベースは、変更するのが容易である。

3　知識ベースンステム開発のライフサイクルに関する要求事項

- 悪構造や変化に富む要求を容易に扱えること。
- 安定したンステムの開発の適合していること。
- ２１６７（Ａ）と一貫性があること。
- スタントアロンの知識ベースンステムばかりでなく組み込み型、リアルタイム型、データトリブン型のンステムに適用できること。

• 要求、設計、インプリメントに対して計画的なレビューや監査をすること。
• 充分かつ最大なＶ＆Ｖをすること。
• メンテナンス業務を拡張すること。

４．完全な機能を満たす知識ベースシステムのライフサイクル（ＦＫＬ）

FKL=Full-function Kbs Life-cycle

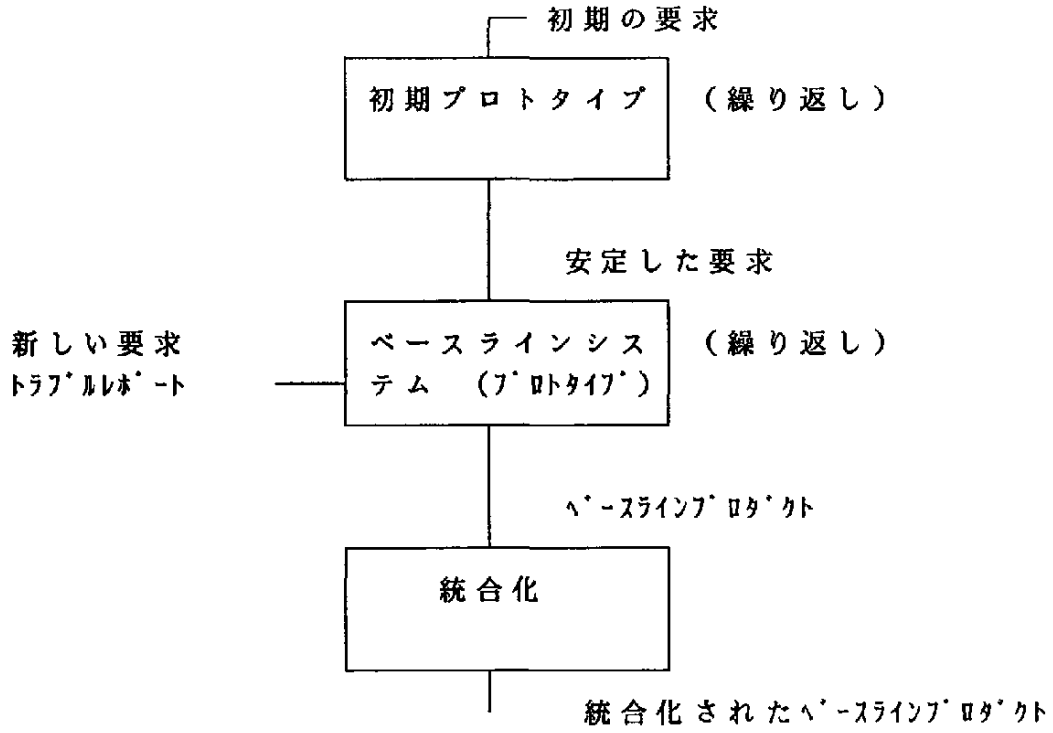


図2.5-2　ＦＫＬのライフサイクル

５．ドキュメント２１６７ＡとＦＫＬドキュメントとの対比

表2.5-3　ドキュメント２１６７ＡとＦＫＬドキュメントとの対比

２１６７Ａドキュメント	ＦＫＬドキュメント
基本ソフトウェア要求仕様書 基本インタフェース要求仕様書	初期ＫＢＳ要求仕様書　(P-KRS)
ソフトウェア基本設計書 インタフェース基本設計書	初期ＫＢＳ設計書　(P-KDD)
ソフトウェア要求仕様書 インタフェース要求仕様書 システム仕様書 システム／セグメント設計書	ＫＢＳ要求仕様書　(KRS)

表2 5-3　トキュメント２１６７ＡとＦ゜ＫＬトキュメントとの対比（つつき）

２１６７Ａトキュメント	ＦＫＬドキュメント
ソフトウェア開発計画書 ソフトウェア試験計画書 ソフトウェア試験手順書	Ｖ＆Ｖ試験計画書 (KDTVV)
ソフトウェア詳細設計書 インタフェース詳細設計書	ＫＢＳ詳細設計書　(KDD)
ソースコードリスト ソースコード 更新済みソースコード ユーザマニュアル プログラマ用マニュアル ﾌｧｰﾑｳｴｱｻﾎﾟｰﾄﾏﾆｭｱﾙ バージョン記述書 ソフトウェア製品仕様書	ＫＢＳｲﾝﾌﾟﾘﾒﾝﾄ書　(KID)
ソフトウェア試験報告書	ＫＢＳのＶ＆Ｖ報告書　(KVVR)
変更の提案（工学上） 仕様変更の注意	ＫＢＳ変更書　(KCD)

６．　知識ベースの適切なテスト法

表2.5-4　知識ベースの適切なテスト法

		テストのタイプ	
		静的テスト	動的テスト
テストの目標	例外	• 例外検出法　(EVA) • 追加した例外(SAIC)	• 構造的テストケースを生成する汎化テスト法　(SAIC)
	不正確 根拠が薄い	• 自動的な知識ベース記述の生成　(SAIC)	• 危機的な欠陥や優先テスト（SAIC） • シナリオテスト法

7．ＦＫＬの開発とテスト計画

- ＦＫＬは複雑で高信頼性のシステムを構築するのに向く。
- テスト
  過去：回帰テスト
  現在：ラムダ機能テスト、強健性テスト
  未来：危機的な失敗の検出

【６】ＥＶＡ　　Rolf Stachowitz氏（ロッキート社）

- Ｖ＆Ｖの評価ツールＥＶＡの目的
  一貫性、完全性、正当性
- ＥＶＡのコンポーネント
  メタ知識ベース、アルゴリズム、トランスレータ（ART,KEE,CLIPS）
- Quintas Prologでインプリメントした。
- 機能
  構造チェッカー
    dead-end、未到着、サイクル、包摂、曖昧性
  ロジックチェッカー
    一貫性、完全性
  拡張構造チェッカー
    ISA,Synoymous,ALLAs
  拡張ロジックチェッカー
    非互換性、IBA、シンタックス、別名
  意味上のチェッカー
    値、クラスの妥当性
- 将来は、要求仕様の妥当性評価をＥＶＡ上で行いたい。

【７】Future Trend in ES V&V

- 複数の専門家の知識獲得と複数の推論エンジンをサポートすへきてある。
- 知識獲得した知識はＶ＆Ｖツールを用いて自動的に評価する機構を取り入れるへきである。
- 経験的知識は、有意義だが複雑であり、クリテカルな応用に向く。

《参考文献》

[Bobrow 86]D.G.Bobrow, M.J.Stefik: Expert Systems : Perils and Promise, Communications of the ACM, Vol.29, No.9, pp.840-894, 1986.

[Buchanan 83]B.Buchanan,et al: Constructing an Expert Systems, F.Hayes-Roth,D.Waterman, and D.Lenat, Addison-Wesley, pp.127-167, 1983.

[Cheng 90]Ana Maria Cheng and Lawrence O.Hall: Expert System Validation as it Applies to Expert Systems using Frame-Based Knowledge Representation, Third Florida AI Reserch Symposium, Cocoa Beach, Florida AI Research Society, St.Petersburg, FL, pp.202-206, 1989.

[Deutsch 81]Michael S.Deutsch: Software Project Verification and Validation, Computer, April 1981.

[Dunn 84]Robert Dunn: Software Defect Removal, McGraw Hill(NY), 1984.

[Gupta 89]U.Gupta: Validation and Verification of Expert Systems, Term Paper, UCF Dept. of Computer Engineering, 1989.

[Hayes-Roth 83]F.Hayes-Roth, D.Waterman, and D.Lenat: Building Expert Systems, Addison-Wesley, 1983.

[Kang 90]Y.Kang and A.Bahill: A Tool for Detecting Expert System Errors ,AI EXPERT, pp.46-51, February 1990.

[Keller 87]Keller: Expert System Technology: Development and Application, Yourdon Press, Englewood Cliffs, NJ, 1987.

[Liebowitz 86]J.Liebowitz: Useful Approach for Evaluating Expert Systems, EXPERT SYSTEMS,Vol.3, No.2, pp.86-96, April 1986.

[Martin 87]N.Martin: The Management of Expert System Development, Tutorial presented at the 3rd Expert Systems in Government Conference, Washington DC., 1987.

[Metzger 81]P.Metzger: Managing a Programing Project, 2nd ed.,Prentice-Hall Englewood Cliffs, NJ, 1981.

[O'Leary 90]T.O'Leary, M.Goul, K.Moffitt, A.E.Radwan: Validating Expert Systems, IEEE Expert, pp.51-57, June 1990.

[Polatikis 85]Peter G.Polatikis: Empirical Analysis for Expert Systems, Pitman Advanced Publishing Program, 1985.

[Ramamoorthy 84]C.V.Ramamoorthy,et al: Software Engineering; Programs and Perspectives, COMPUTER, pp.191-209, October 1984.

[Smith 90]Smith: Validation of Expert Systems, FLAIRS-90, pp.197-201, Cocoa Beach, Florida, 1990.

付録B．エキスパートシステム評価の文献調査

海外におけるエキスパートシステムの評価に関する動向を把握するため、幅広く文献の収集を行った。その中から、本プロジェクトで特に有用と判断した１４文献について詳細な調査を行ったので、その内容を以下に紹介する。

なお文献の分類は下記の項目に従う。

- 方法論的基盤（Methodological Basis）
- 検証（Verification）
- 妥当性検査（Validation）
- 信用性・評価（Credibility and Evaluation）
- 手法・技法（Methods and Techniques）
- 統計手法（Statistical Tools）
- 自動化（Automation）
- 事例（Experience）
- ライフサイクル（Life Cycle Issues）
- その他（Others ）

1. エキスパートシステムの失敗の原因

タイトル：	Why Expert Systems Fail
著　者　：	BELL,M.Z.
出　典　：	Journal of the Operational Research Society, Vol.36,No 7,pp 613-619,(1985)
分　類　：	Methodological Basis
ｷｰﾜｰﾄﾞ　：	開発フェーズ、失敗の原因

（１）概　要

エキスパートシステム開発が失敗する原因のいくつかは従来型のシステム開発の失敗の原因と同じものであるが、次のようなエキスパートシステムに固有の問題もある。

- 対象としている問題が理解できない、あるいはされないものである。
- 常識、知識の限界についての知識、が必要とされる。
- テストができない。
- 信頼されない。

（２）はじめに

エキスパートシステムの開発はフェーズで明確に分けられていると言うよりは、インクリメンタルな開発手順になるが、内容的には通常のソフトウェア開発と同様以下の６つのフェーズを含む。

①分析（analysis）
②設計（design）
③開発（implementation）
④テスト（testing）
⑤ユーザの受け入れ（user acceptance）
⑥保守（maintenance）

以下で各フェーズごとに失敗の原因を検討する。

（３）問題の分析と知識の抽出

専門家から必要な知識（expertise）を得ることがエキスパートシステム開発の必要条件である。ここでの問題は以下の通りである。

①専門家が多忙で時間がさけない。

知識獲得に多大の時間を要すため、多忙な専門家が協力するのは無理である。

②専門家が彼のアイディアを説明できない。

専門知識の内容を説明するのは難しい。専門家自身も彼が得た結論の道筋を理解していなかったり、実際に用いている問題解決方法とは異なった説明をすることもある。

③専門家が知識抽出に協力的でない。

専門家がエキスパートシステム構築に反対だとすると、知識は得られない。このため、専門家は開発者と同じ立場に立ってもらうことが必要である。

④そもそも専門家が存在しない。
必要な知識が存在しない分野もある。音声認識などがその例で、我々すへてが音声認識の専門家とも言えるが、だからと言ってマイクロフォンからの波形を見て我々に音声認識のプロセスが説明できるわけではない。

（４）設計と開発
数少ないケースに対してのみ有効であるようなシステムは簡単に構築てきるが、多くのケースに有効なシステムの開発は難しく、さらに、想定されるほとんどのケースを扱うことができるようなンステムを構築することはほとんと不可能である。

（５）常識と閉世界仮説
常識の欠如はエキスパートシステムの大きな問題である。エキスパートシステムは事前に言われたことだけを知っているが、全ての事をあらかしめ言っておくことは不可能である。

（６）知識表現の選択
知識表現の選択に関する中心的な問題は以下の３点てある。
①知識表現言語の機能が十分でない。
②知識表現言語の機能が十分すぎる。
③知識表現言語の形式が専門家の知識の形式に合わない。
対象としている問題の特性に合致した知識表現手法を選択することが重要である。

（７）テスト
エキスパートンステムのテストが困難なのは以下の理由による。
①システムが異常な事象を扱うように設計されていること。
現実に発生させにくい異常事象はシミュレーションて検証するしかない。しかしシミュレーション時にエキスパートシステムを構築したのと同じ誤った仮定をしていない保証はない。
②システムがカバーしている領域が広すぎること。
③システムのテストに費用がかかりすぎること。
石油市場の取り引きシステムを開発したとする。この場合売り買いの結果が市場そのものに影響するので、実際に売買してみないとンステムのテストができないが、失敗した場合あまりにも高くつく。
④システムのテストに時間がかかりすぎること。
適切なテストケースを生成するだけて多大の時間を要することがある。
⑤正しい解が何かわからないこと。
医療の診断問題などでは何が正しいかについては専門家の間でも意見が別れることがある。

以上のような場合は何れにせよ網羅的なテストは不可能なので、何らかの妥協が必要である。これについては次のような手法がある。
①システム構築のためのデータとテストのためのデータを分離する。
システムのテストに使えるデータのうちの90パーセントをシステム開発に

用い、残りの10パーセントはテストのためにとっておき、それでンステムのエラーの頻度を見積もる。全てのデータを用いないことによってンステムの性能が落ちることがあり得るが、実際にシステムがとの程度使えるかを知るためにはこの方が良い。

②システム開発者とテストのためのデータ作成者を別にする。

システム開発時とテストデータ作成時に同じ誤った前提を持ち込むことを避けるために、この2つを異なったグループで実行した方がよい。

③システムの取った行動の正当性を別の専門家が判断する。

システムに知識を提供した専門家と異なった専門家が結果の正当性を判断する。ただし、これはシステムが正しいことを示すのではなく、単にもっともらしいということを示すものである。

（8）ユーザの受け入れ

ユーザの受け入れ時の評価はンステムの性能の評価と異なり、次のような人間的な要素が客観的な評価を妨げる。

①信頼の欠如

システムはエントユーザからの信頼を得なければならない。このためには説明機能が重要であるが、リアルタイムンステムなどの場合、処理の時間的な制約によって説明が不要、不可能な場合もある。ユーザと協調して問題解決に当たることができるような場合にはシステムが常に完璧である必要はない。

②信頼しすぎ

一般に人々はコンピュータから出てくる結果を過度に信頼する傾向がある。ユーザは本来エキスパートシステムにもできる仕事の一部を負担すへきたし、こうすることによってコンピュータを過度に信頼しすぎる危険を軽減することができる。

③法律上の責任

エキスパートシステムの結果によって誰かが被害を受けたり、損害を被った場合の責任は誰にあるのだろうか。法律上の責任は特に医療ンステムの場合に大きな問題を生ずる。

④仕事の満足度

誰も自分の仕事がコンピュータで置き換えられたり、また仕事の重要性や面白さが減しられたりすることを好まない。ンステムは人間に替わることではなく、彼を助けるものてなければならない。そうすることによって、人間を「単なる機械には難しすぎる」問題に専念てきるようにするものてなければならない。

（9）保　守

- 知識の追加は大幅な書き直しを意味するのだろうか。
- 所得税のためのンステムに対して、所得税法の変更をしたらとうなるたろうか。
- システムはどのくらいの間にユーザに最新の機能を提供することができるだろうか。

（１０）結　論

エキスパートシステムの開発で従来型のシステム開発と異なる困難な点は知識の抽出に関わる問題である。常識のからむ誤りを避け、思った通りの動作をするように知識を抽出しなければならない。

一番の問題は閉世界仮説の問題である。システムは自分が知らないことが何かを知らなければならない。少しの知識は、ある意味では全然知識を持たないことより危険である。

（１１）おわりに

エキスパートシステムの分野はまだ始まったばかりである。したがって開発されたすべてのエキスパートシステムが現場で実際に用いられると期待するのは現実的ではない。多くの開発者がこれまでしてきたことに対するより良い方法も見つけられているので、今後はさらに良いエキスパートシステムも開発されるだろう。

2. 原子力発電向けエキスパートシステム開発におけるV&V方法論

タイトル：	Verification and Validation of Expert Systems for Nuclear Power Plant Applications.
著　者　：	Naser, J. A.
出　典　：	EPRI Report, No. NP-5978, 116 pp ,August 1988
分　類　：	Methodological Bases, Life Cycle Issues
ｷｰﾜｰﾄﾞ　：	Software V&V, Expert Systems, Nuclear Power Plants, Life Cycle, EOP Tracking System, REALM Expert System

（１）概　要

本報告では原子力発電所向けエキスパートシステム開発におけるＶ＆Ｖ方法論について述べる。報告の基本的な立場は、エキスパートシステムもソフトウェアシステムであるから、明確なライフサイクルをもち（従来の手法を含む）Ｖ＆Ｖ手続きを適用できるというものである。

まずエキスパートシステムと従来型システムの類似点・相違点について分析を行い、従来のＶ＆Ｖ方法論が適用できる範囲を明らかにする。次に、従来のＶ＆Ｖ手法で対応できない場合に使う、エキスパートシステムに特化したＶ＆Ｖ手法を示す。これと従来のＶ＆Ｖ方法論とを統合することで原子力分野向けのＶ＆Ｖ方法論が構成される。さらに、エキスパートシステムをＶ＆Ｖを行う際の難しさに応じて６種類のタイプ分類し、各タイプに合ったＶ＆Ｖのガイドラインを示す。

実際の適用にあたっては、この方法論を個々の問題向きに特化することも必要である。そこで、EOP (Emergency Operating Procedures) Tracking System とREALM (Reactor Emergency Action Level Monitor) という２つのエキスパートシステムに対して、このＶ＆Ｖ手法を適用した結果を報告する。

（２）序　論

Verificationとは、システム内部の正しさを保証することである。すなわち、開発者のプランを詳細化、プログラム化するときに生ずるエラーをなくす作業である。一方、Validationとは、システムを外部からみたときの正しさをさす。すなわち、システムが現場で運用されているときに正しい望みの結果がでるようにするための確認作業である。

エキスパートシステム技術はまだ新しく、特に原子力分野の応用は少ない。Ｖ＆Ｖ問題もまだ研究課題である。しかし、エキスパートシステムは、従来システムと異なる面もあるが、似た面もあり、したがって、従来システムのＶ＆Ｖ技術と組合せることによって成果を上げられる。

（３）ソフトウェアのＶ＆Ｖに対する従来の接近法

従来型ソフトウェアのＶ＆Ｖ手法は確立されており、そのためのツールや方法論も存在する。従来型システムは Waterfall型のトップダウン的方法で開発されることが多く、その手法は、Constantine, Yourdon, DeMarco 等によって構造化設計法として発展してきた。Ｖ＆Ｖもソフトウェア開発のライフサイ

クルに組込まれており、基本的に、要求を明確に定義し、それに対してV&Vを行うこととなる。

（4）エキスパートシステムの概要

エキスパートシステムは、知識ベース、データベース、推論機構、制御機構、利用者インタフェースから構成される。システムの開発はンステムを既存のツールを使って開発する場合は普通、推論機構、制御機構は Validate されたものとするが、カスタマイズした場合はそうではない。利用者インタフェースに対するV&Vは従来の方法で行うことができる。知識ベースを評価するには、Accuracy, Logical Consistency, Logical Completeness, Functional Completeness をチェックしなければならない。

多くのエキスパートシステムは図1に示す6つのタイプにわけることができる。

	エキスパートンステムのタイプ					
ライフサイクル	1 単純 成文化	2 単純 成文化 不確か	3 単純 専門知識の獲得	4 単純 専門知識の獲得 不確か	5 複雑 専門知識の獲得	6 複雑 専門知識の獲得 不確か
要求分析	全テスト生成		いくつかのテスト生成			
	V&V レビューの要求					
	V&Vの計画とスケジュール					
設計プロトタイピング	ツール選択の妥当性				V&V推論エンンンの設計	
	V&V 不確かさの操作					
	V&Vレビュー要求					
	KBテストの残りを生成					
	各プロトタイプのV&Vレビューの設計					

図1(a) 各エキスパートンステムのタイプとV&Vライフサイクル

	エキスパートシステムのタイプ					
ライフサイクル	1 単純 成文化	2 単純 成文化 不確か	3 単純 専門知識の獲得	4 単純 専門知識の獲得 不確か	5 複雑 専門知識の獲得	6 複雑 専門知識の獲得 不確か
プログラミングテスト	KBコーデングの妥当性		KBのV&V			
	ツールなしでソフトのテスト				全てのソフトをテスト	
	テストの不確かさ					
	システムとユーザの評価					
	構成管理の確立					
インストール運用とメンテナンス	受け入れテストとインストール					
	最終的なV&Vレポート					
	KBの修正とテスト					
	構成管理の継続					

図1(b) 各エキスパートシステムのタイプとV&Vライフサイクル

(5) オブジェクト指向プログラミングとエキスパートシステムのV&Vとの関連

オブジェクト指向プログラミングによってンステムを開発した場合、

①オブジェクトで表現した知識と要求定義との対応がとりやすいためにValidationが容易になる。

②ルール数が少なくなる。

③継承機能を使うことでプログラミング作業が軽減される。

④モジュラリティが向上する。

⑤従来型ンステムのテスト手法が採用しやすくなる。

という利点が生ずる。

(6) エキスパートンステム V&V手法の選択

エキスパートンステムは通常段階的な方法で開発されるために、開発初期段

階では要求仕様が明確でない。しかし、V&Vを開発のライフサイクルに組込み、開発の各段階ごとに要求定義が定まった時点で、それをテストに対する基準としていくことが必要である。従来型システムのV&V手法も使える部分には併用すべきである。また、何を何故テストするのか、誰がテストを行うのか管理するのかを明確にしておくことも重要である。

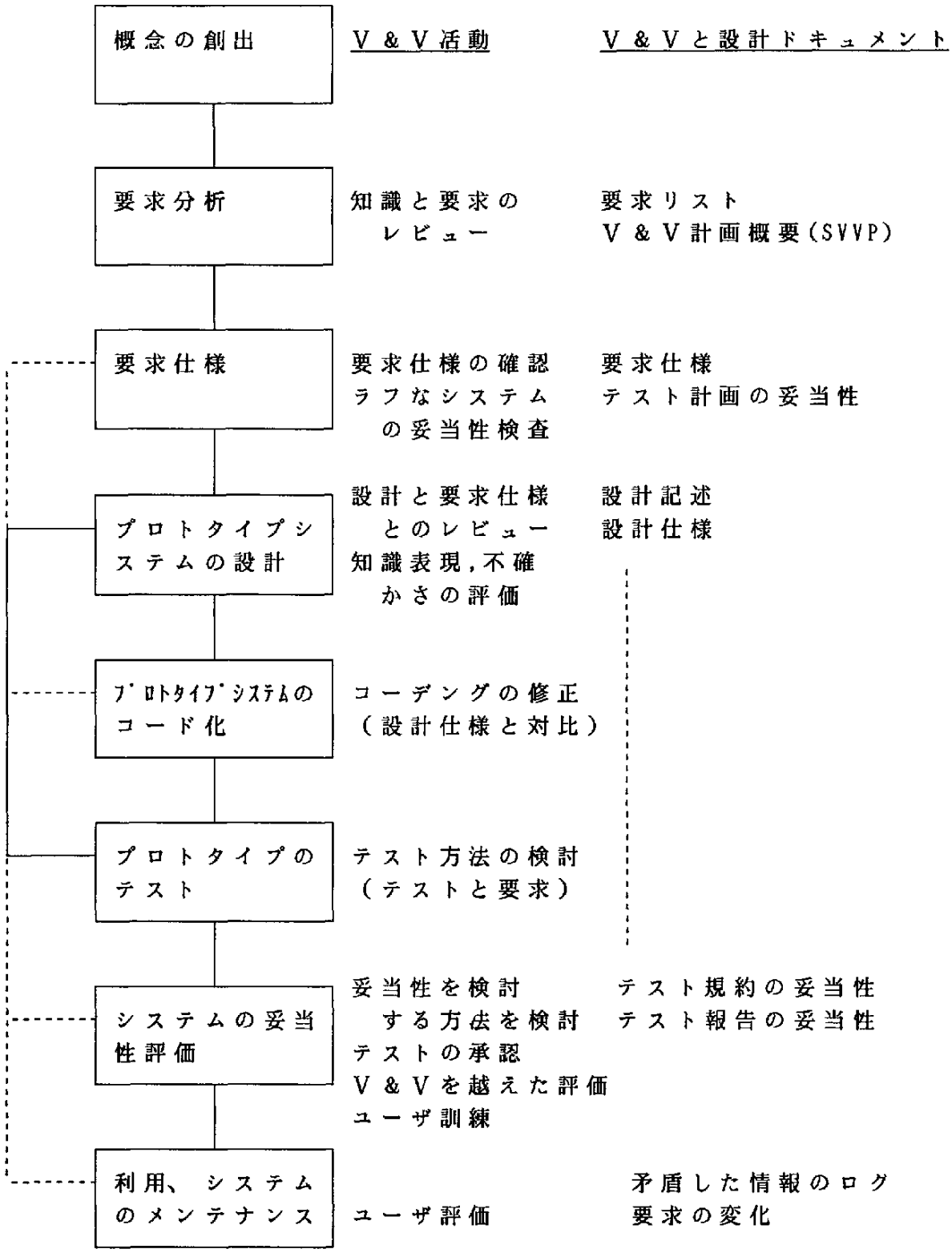
（7）知識ベースのテスト

エキスパートシステム全体のテストを完全に実施することは困難であるが、知識ベースのテストとはその正確さと適切さを調べることである。このためには、以下のような項目がある。

①知識表現の適切さ
②知識の正確さ（専門家の知識との比較；根本的な基準との比較）
③ルールベースの一貫性と完全性
④推論方式の正しさ

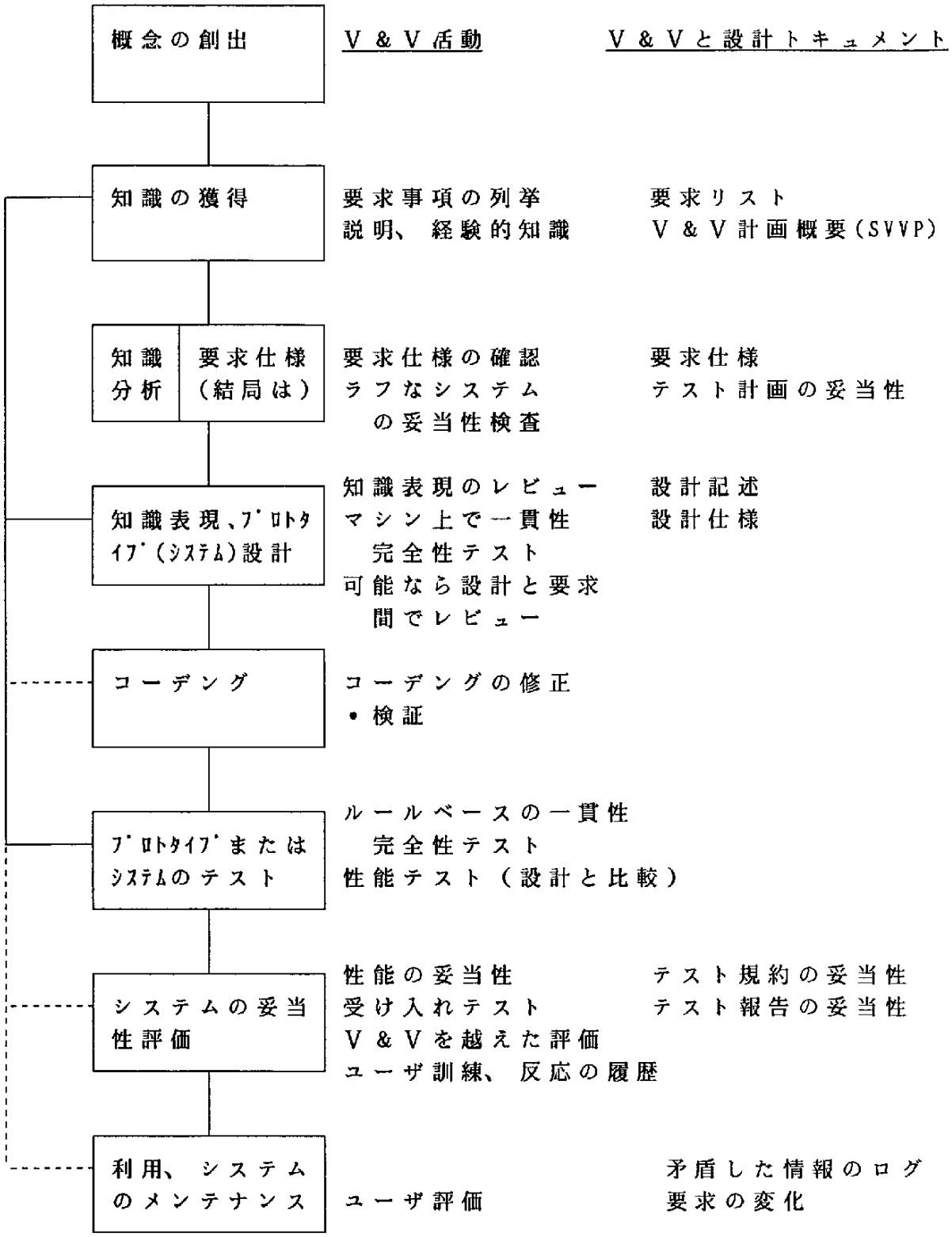
（8）エキスパートシステムのV&V計画

タイプ1，2とタイプ3，4のエキスパートシステムに対しては、図2のようなV&Vの計画を立てることができる。しかしタイプ5，6のエキスパートシステムに対しては、このような明確な計画はたてることが難しい。



(a) エキスパートシステム(タイプ1、2)のライフサイクル
(妥当性ある成文化された知識が既にあるとき)

図2 エキスパートシステムのV & V(その1)



(b) エキスパートシステム(タイプ3&4)のライフサイクル
(知識獲得を行うとき)

図2 エキスパートシステムのV&V(その2)

（９）EOP とREALM に対するＶ＆Ｖ

EOP (Emergency Operating Procedures) Tracking System は、緊急時の原子力プラントの状況をリアルタイムで分析し、適切な緊急時操作手順を運転員にその根拠とともに提示するエキスパートシステムである。このエキスパートシステムはルールで記述され、タイプ１に属している。そのＶ＆Ｖには、EOP に記述されている知識が正しく知識ベースに投入されているかどうかを確認すればよい。

REALM (Reactor Emergency Action Level Monitor)は、原子力プラントの異常事態を同定し、その重要度を評価するreal-time Expert Assistance である。REALM はタイプ３のシステムで、ルールベース推論とオブジェクト指向プログラミングとを用いて実現されている。その知識ベースは、EAL (Emergency Action Level)という記号化された知識と専門家から獲得した知識とからなり、複数のコンポネントから構成されている。そのためこのシステムのＶ＆Ｖ手法は非常に困難である。ここでは個々のコンポネントごとにＶ＆Ｖを実施した。

（１０）結　論

本報告では原子力発電所向けエキスパートシステム開発におけるＶ＆Ｖ方法論について述べた。そのポイントは従来のＶ＆Ｖ方法論にエキスパートシステム独特の手法を組合せることであった。エキスパートシステムのＶ＆Ｖはまだ新しいテーマであり、今後の技術の進展に期待したい。

3. 知識ベースシステムのV&Vアプローチ

タイトル：	A Comprehensive Approach to the Verification and Validation of Knowledge-Based Systems
著　者　：	Miller,L.A.
出　典　：	Presentation at the Workshop on Verification, Validation and Testing of Knowledge-Based Systems, held at the IJCAI'89 ,Detroit,Michigan (19 August, 1989)
分　類　：	Methodological Basis, Life Cycle Issues
キーワード：	V&V, Testing Criteria, Testing Method, Life-Cycle

（１）はじめに

本稿では、知識ベースシステムの実用的なＶ＆Ｖ（Verification and Validation）のアプローチを述べる。Ｖ＆Ｖの定義はIEEE729-1983によると次の通りである。

Verification（検証）とは、開発フェーズのプロダクトを決定するプロセスが、前の開発フェーズで確定した全ての要求を満たしているかである。
Validation（正当性）とは、ソフトウェアの要求に対して最終的に承諾を得るプロセスのことである。

以下に述べる点は、長年かけて得られたＶ＆Ｖに対する知見である。

① 知識ベースの安全性、信頼性の問題はＶ＆Ｖに対する要求の強さに比例して対処すべきである。特に、組み込みシステム、リアルタイムシステム、センサ入力のシステム、他の複数のソフトウェアとインタフェースを持つ場合は、積極的にＶ＆Ｖを行う必要がある。

② ただし、Ｖ＆Ｖを緻密に行うとプロジェクト予算の半分を使うケースがあるなど、かなりのコストがかかるので、実際上は、出来なかったＶ＆Ｖに対してどう保護するのかが問題となる。

③ ＡＩの開発者は通常ハッカーのような態度であり、従来のソフトウェア工学を生かしたよいプログラミングには耳を傾けたがらないが、知識ベースシステムの開発ツールに対しては大変興味を持っている。

④ 開発システムとデリバリシステムとのギャップを小さくするよう努めなさい。この間の差があれば最終版のシステムとすべくメンテナンスをしなさい。

⑤ テストは、最も問題の起こしがちな他のソフトウェアとのインタフェースに重点を置くべきである。様々なユーザの要求や知識ベースにはそれなりの方法を必要とするから、テスト方法だけがベストのＶ＆Ｖではない。

以下に、知識ベースシステムの複雑さ、知識ベースシステムの要素、知識ベースシステムのＶ＆Ｖのタイプ、テスト基準、テスト方法の適用、ライフサイクルについて述べる。

（２）知識ベースシステムの複雑さ

表１では、知識ベースシステムの複雑さを、探索空間、不確かさ、決定プロセス、知識源、他システムとの連動、の５つの要因で示している。この要因の

並び順序は、開発の困難さが増すに連れ上から下にいく。

表１　知識ベースの複雑さによる分類

	単純なシステム	複雑なシステム
探索空間	• 単純／小さい • 大きいが探索が簡単	• かなり大きい • 不確定の要因あり
不確かさ	• なし	• 曖昧な推論 • 雑音、ﾐｽﾃﾞｰﾀ、不確なデータによる推論
決定プロセス	• 後向き推論 • 前向き推論	• ＴＭＳ • 制約指向推論 • 組み合せ
知識源	• 明確な要求 • 成文化された知識	• 曖昧な要求 • 複数の知識源の引出し
他システムとの連動	• スタンドアロン型 • コンソールインタフェース	• 組み込みシステム • ﾃﾞｰﾀｲﾝﾀﾌｪｰｽ • ﾘｱﾙﾀｲﾑｼｽﾃﾑ • 割り込み型

（３）知識ベースンステムの要素

知識ベースシステムは、知識ベース、シェル、推論機構、外部インタフェースの４つの要素で構成される。この各要素のサブ要素は、表２の通りである。サブ要素をさらに手続き型（Ｐ）、事実型（Ｆ）、ルール型（Ｒ）の３つに分類する。

（４）知識ベースシステムのＶ＆Ｖのタイプ

Ｖ＆Ｖのタイプは以下のように分けられる。

- 手続きベースのＶ＆Ｖ
- 宣言ベースのＶ＆Ｖ ┬ ルールベースのＶ＆Ｖ
  └ 事実ベースのＶ＆Ｖ

（５）テスト基準

テスト基準を、機能基準、性能基準、一般的な基準の３つに分類し、その詳細な基準を、表３に示す。

表２　知識ベースシステムの要素

主要素	サブ要素	分類
知識ベース	ルールベース	R
	事実	F
	連想データベース	F
	フレーム	F
	デーモン	P
	手続き、機能	P
シェル (外部ｲﾝﾀﾌｪｰｽのｻﾌﾞｾｯﾄ)	ユーティリティ	P
	ユーザインタフェース	P
	知識表現	P
推論機構	パターンマッチャ	P
	決定手続き	P
	ルールの順序制御	P
外部インタフェース	ＯＳ	P
	データベース管理システム	P
	Ｉ／Ｏデバイス(ﾕｰｻﾞｲﾝﾀﾌｪｰｽ)	P
	ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ言語	P
	他のアプリケーション	P
	ﾕｰﾃｨﾘﾃｨ、機能、ﾌﾟﾛｼｼﾞｬｺｰﾙ	P

表3　テスト基準

分類	基準	分類	基準
機能基準	一貫性 完全性／網羅性 正確性 解の個数 説明 メタ知識	一般的な基準	理解性 交信性 アクセスコントロール セキュリティ 安全性 ﾌｫｰﾙﾄﾄﾚﾗﾝﾄ/ｴﾗｰﾊﾝﾄﾞﾘﾝｸﾞ 頑健性 変更可能性/柔軟性/ 拡張性/増大性 自己記述性 トキュメントの適切さ コスト 簡潔性 装置独立性 モジュラリティ 相互運用性 通信基準 読解性 データ基準 再使用性 単純性 構造性 テスト可能性 追跡可能性
性能基準	実行速度 メモリ要求 Ｉ／Ｏハントリンリング 性能（専門家が問題解決に要する時間，質，成功回数，経験のレベルとの対比） 効率性 ﾃﾞｰﾀﾍﾞｰｽのｱｸｾｽﾀｲﾑ 解決までに要する時間 精密さ		
一般的な基準	移植性 保守の良さ 信頼性 可用性 利便性 運用可能性 ヘルプ／指導性		

（６）テスト方法の適用

テスト方法は、動的テストと静的テストに分類される。

a．動的テスト

①ランダムテスト（Random test generation）

ランダムにテストデータまたはテスト項目を選択してテストすることである。

②機能テスト（Functional Testing）

ンステムの機能要求仕様書に基づいて、各機能についてテストすることである。俗にブラノクボノクステストという。

③構造テスト（Structual Testing）

ソフトウェアの構造を考慮して、主として制御構造やパスを対象にテストする。俗にホワイトボノクステストという。

④回帰テスト（Regression Testing）
システムまたはシステム構成要素の変更中に生じた欠陥を検出し、意図しない有害な影響が生じていないことを、または修正後のンステムまたはシステム構成要素が、その規定された要求事項に合っていることを検証するために再試験することである。
⑤説明にもとづくテスト（Explanation）
知識ベースシステムを実行して、ルールをトレースし付随して出力される説明に基づいて、ルールの実行順序がよいかどうかを調へることである。
⑥シミュレーション（Simulation）
入力データのシミュレータはできるだけ現実の入力パターンを再現して生成する。特に組み込み型リアルタイムシステムでは有用である。

b．静的テスト
①異常検出（Anomaly Detection）
ルールの異常検出には、構造的な異常検出と意味的な異常検出がある。
構造的な異常検出には、行き止まりノード、非参照な事実、重複ルール、包摂、クローズ内直接サイクル、クローズ間間接サイクル、直接的にわかる矛盾、間接的にわかる矛盾、関係ない述語の挿入がある。
意味的な異常検出には、不備なルール、事実の範囲の誤り、誤りのある値、非互換の値、対象が誤りのある部分的な関係がある。
②構造的ウォークスルー（Structured Walk Throughs）
開発チーム内の他の要員が書いた設計書、コートを通して、技法、仕事のやり方、犯しやすい誤り、開発規格に対する違反やその他問題について質問、意見を述べ再検討する過程である。
③非公式なパネル形式による検査（Informed Panel Inspection）
そのシステムが採用した推論機構について専門家を集めてパネルディスカッション形式で検査することである。
④クリーンルーム方式による検査（Clean Room Inspection）
今までは、開発者がデバックをしてプログラムを完成したが、本手法は開発者にはプログラムを実行してデバックすることを禁止して、静的な検査手法を用いてプログラムを検査することである。クリーンルームとは、半導体製造工場からきており、最初からゴミを入れないことを意味している。
⑤数学的検証（Mathematical Verification）
プログラムがその仕様を満たすことを数学的に証明する形式的技法である。例えば、各プログラムにASSERTION文を入れてプログラムの定理証明を行うことである。
⑥記号を用いた実行（Symbolic Execution）
実際の値を入力データとするのでなく、記号を入力してプログラムをシュミュレートし、出力はこれらの記号を含んだ論理式または数式として表す検証の技法である。
⑦フレーム表現が良形式かの検査（Well-formedness）
良形式であれば知識の妥当性が増加し、知識の適用する範囲がより広くなるので良形式かどうかを検査することである。
⑧故障の木解析（Fault-Tree Analysis: FTA）
システムの致命的故障あるいはシステムの使命達成上その発散防止したい、

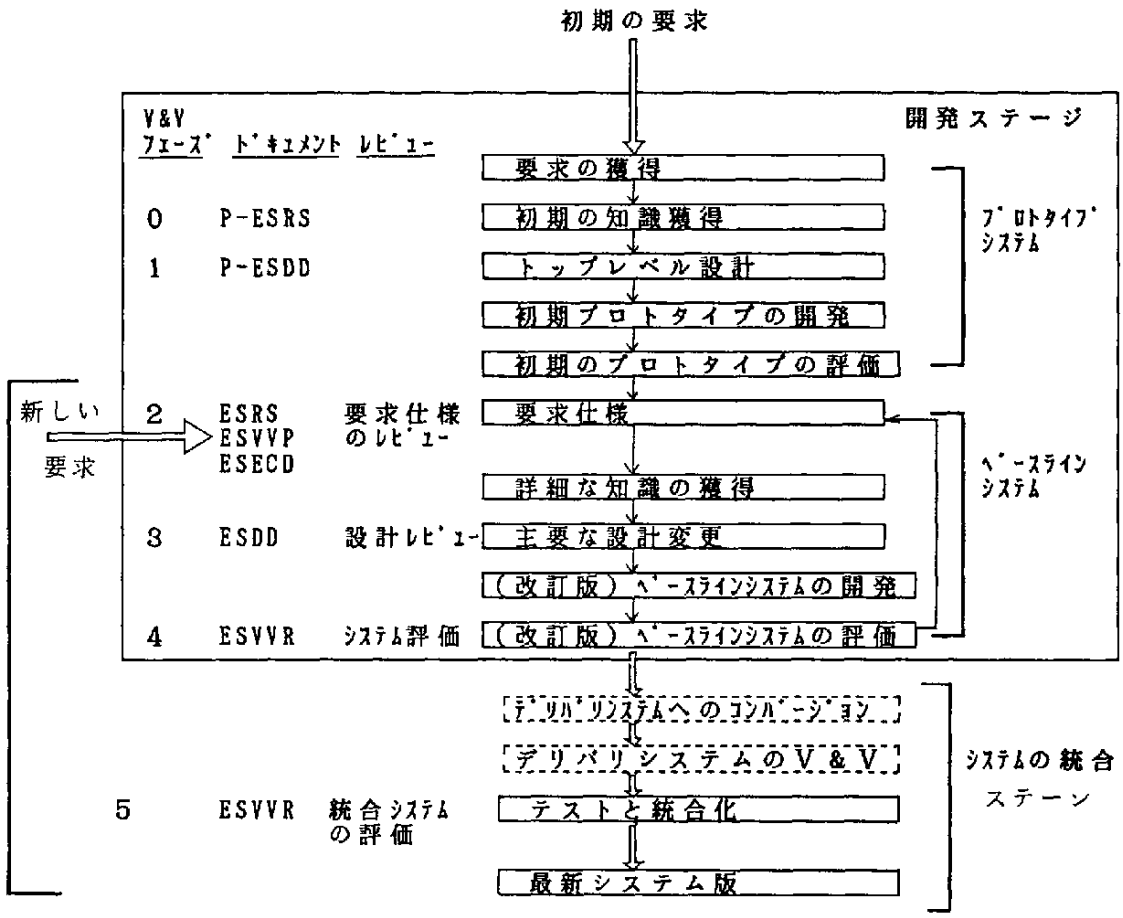
“好ましくない事象”を検出する図式解析法である。

⑨自動データフロー解析（Automated Data-Flow Analysis）

この方法は、ある特定のデータに着目して論理的なシーケンスに基づいて実行する。データフロー解析をすれば、着目する変数がどこで定義され、どこで参照され、どの時点で値が未定義かがわかる。

（７）知識ベースシステムのライフサイクル

知識ベースシステムのライフサイクルを図１に示す。



注）P-ESRS:初期の要求仕様
P-ESDD:初期の設計ドキュメント
ESRS　:要求仕様
ESVVP :Ｖ＆Ｖ計画
ESECD :変更ドキュメント
ESDD　:設計ドキュメント
ESVVR :Ｖ＆Ｖ報告書

図１　知識ベースシステムのライフサイクル（提案）

4. エキスパートシステムの検証と妥当性検査

タイトル： Verification and Validation of Expert Systems
著　者　： Geissman,J R. and Schultz,R D
出　典　： AI EXPERT,Vol 3,No 2,pp 26-33,(Feb 1988)
分　類　： Validation, Verification, Life Cycle Issues
ｷｰﾜｰﾄﾞ　： チェックリスト、ライフサイクル

（１）Ｖ＆Ｖとは

エキスパートシステムが使われるためには検証(Verification)と妥当性検査(Validation)が必要である。検証とはエキスパートシステムが正しく開発され、技術的なエラーがないか調へることを意味し、妥当性検査とはエキスパートシステムがユーザのニーズを満足することを保証することである。検証では上位レベルの設計が要求仕様の全ての項目を取り込んでいるか、詳細設計の中で記述されているように動くかを調へ、妥当性検査では完成されたプログラムが要求仕様書の中の機能を満足するかどうか、意図された目的のために使えるかとうかを検査する。

（２）開発局面

エキスパートシステムの開発には４つの局面がある。

①第１局面：問題の決定

大きな目的はプロジェクトが真の要求を満たし、技術的に可能であることを保証することである。

②第２局面：初期プロトタイプ

初期プロトタイプの段階での大きな目的は拡張されたエキスパートシステムの技術的、経済的な可能性をすばやく示すことてある。

③第３局面：拡張プロトタイプ

初期プロトタイプと計画の通りの拡張プロトタイプの目的は動作の範囲として問題の複雑さ全部を扱えるところにまで広げることてある。

④第４局面：製品版システム

製品版システムの目的は実行運用環境（ＰＣ，メインフレームなと）の範囲をカバーすること、重要なシステムの機能やメモリ使用なとに関して最適化することである。プロトタイピング局面が明確にたとれるように記述されていたら、要求をまとめた仕様と知識ベース設計のトキュメントによって、製品版システムを検証することができる。

（３）エキスパートシステムの要素

エキスパートシステムには３つの処理モジュールが入っている。それは、推論エンジン、説明機能、そしてユーザフレントリなインタフェースてある。その他に知識モジュール（データベースと経験的知識が格納されている）、作業領域用のダイナミックメモリモジュールがある。

知識ベースにはある問題をどのようにして解くかいうことと、考慮しなくてはならない重要な事実の記述が入っている。知識は特有の知識表現て表現され

る。例えば、知識はIF-THEN ルール、事実、ゴールといった形式で書くことができる。

作業領域には、現在取り組まれてる問題に関する記述が入っている。推論エンジンは作業領域からの情報を解釈し理解するために、知識ベースから手続きを選択する。ユーザインタフェースによってユーザに入力情報を要求し、結論を表示させるという対話ができる。

（４）Ｖ＆Ｖを実施するための問題点

Ｖ＆Ｖを実施する際に、以下のような問題点がててくる。

①エキスパートシステムの開発計画が曖昧である。

曖昧な目的のまま開発を開始すると、「何もないよりはましだから」といって、「実際にシステムが何をするのかはどうでもいい」という結論に達することがある。この考え方にもそれなりの理はあるだろうが、ンステムは何をするのかということが重要である。

②テスト可能な要求仕様を見つけることは難しい。

エキスパートシステムは非手続き的なアーキテクチャであるため、実行の流れに類似した形でのコードはつくらない。それゆえ、コートを試験して実行をトレースする技術は推論エンジン自体にはうまく当てはまるとしても非手続き的なアーキテクチャには当てはめることができない。

（５）６段階のＶ＆Ｖ方法論

上記のような問題を解決しＶ＆Ｖを行なうため、以下に６段階のＶ＆Ｖ方法論を示す。

［第１段階］初期プロトタイプを開発して、テスト可能な要求仕様を作成すること。

要求仕様及びプロトタイプは、プロトタイピンクを繰り返すことによってスパイラルモデルと似たような発展をする。

［第２段階］明確に定義された問題解決戦略(Paradigm)の用語を使って設計すること。

コードレベルでできあがったンステムは、意味のある解析可能な構造をしているのかを確かめる。そのため既に確認されたパラダイムたけを用いて設計する。

エキスパートシステムにおける「構造化設計」とは、以下のことを意味する。

- １つかそれ以上の知識処理パラダイムを正式に定義すること。
- 妥当で保証されたパラダイムを限定されたセットのみてしか使わないこと。
- 例外的に取り入れた全ての手続き型のコートに対して独立にＶ＆Ｖを行うこと。

［第３段階］推論エンジンの正しさを証明すること。

推論エンジンの証明とは、推論エンジンが第２段階て規定した１つまたはそれ以上の知識処理パラダイムを実行するかどうかをテストする。

推論エンジンの保証とは次の通りである。

- 推論パラダイムを形式的に定義すること（ＡＮＳＩ規格のように）。
- 特定の推論パラダイムに関して、特定のエキスパートンステムツールの用語を用いて、どう表現されているかを仕様化すること。
- 抽象的に表現されたパラダイムに適したテストの仕方を決めること。
- 特定のツールの言語に適するようにテストを翻訳すること。
- 特定のパラダイムを保証するため、テストを実施すること。
- （満足されたルールのリスト、作業用メモリの内容、継承ネットワーク、最新のゴールといった）推論エンジンによって与えられる、全てのデバッグ用または説明の情報が、正しいかを検査して、後の検証の局面においてもあてが出来るようにすること。

［第４段階］検証を容易にする設計を行うこと。

エキスパートシステムがどのように知識を使うべきであるか、そして、プロトタイプの抽象概念にどのように基づくかという高位レベルの記述を行なう。

エキスパートシステムの上位レベルの設計の重要な要素は以下の通りである。

- 個々の要求をどのように扱うつもりでいるか。
- フォローすべき知識処理パラダイムを決定する（これらは保証された推論エンジンにサポートされたものの中から選ばれるべきである）。
- 信頼して処理を行える原理的な事実に関する知識を用意する。
- 全ての解法を小問題に分割し、１つの小問題から次の問題に移り、小問題同志のコミュニケーション、そしてこの構造がどのように表現するか（例えば、後ろ向き探索システムにおけるゴールのツリー）の方法を設計する。
- 必要なデータをどのように得るつもりでいるか（ユーザへの問い合わせ、デバイスからの読みとり、対話型データベースシステムを含む）をも含めた外界との全ての相互作用を設計する。
- 解法がより所としている基本的な諸仮定、それが妥当である範囲、境界条件を設計する（この情報は、例えば解析的な仮定が妥当である条件や入力の範囲といった手続き型言語の正しさを証明するのを助ける不変量の主張に相当する）。
- エキスパートシステムの中で、問題の現在の状態、すなわち、世界の状態が作業用メモリの中で一般的な用語でどのように表現するかを設計する。

［第５段階］知識ベースを検証すること。

知識ベースを以下のチックリストを用いて確認する。

- 知識ベースが保証されたパラダイムを遵守しているかどうかを確かめよ。
- 小問題の構造を検証せよ。そして各々の小問題が互いに独立であるかを検証せよ。
- 静的、個別的な意味で知識が正しい、または適切であることを確かめよ。
- ＬＩＳＰ、Ｃ、データベースシステムを例外的に呼び出すようなパラダイムの要素でない知識ベースの部分を識別せよ。
- 不変量と境界条件が成り立たなくなったときに知らせてくれる組み込まれた付加的なデモンルールを確かめよ。

［第６段階］正式な妥当性検査を行うこと。

妥当性検査の尺度を決定し、客観的にそれらの度合を決定する。

この妥当性検証のステップは以下の通りである。

- 妥当性検証の評価基準に対する尺度を決定せよ。
- 選ばれた評価基準に対する尺度の客観的な測定項目あるいは代用特性を決定せよ。
- エキスパートシステムが確実に正しく扱う必要のあるデータの集合の範囲を規定せよ。
- 当該問題に特化したテストケースのライブラリ及びシナリオとを開発し、知識ベースが修正された時に回帰テストを行え。ソフトウエアのテストと同様に試験者は正常な状況に対して適切な反応をするのかを確認し、システムエラーを引き起こさせるようにすべきである（良いテストとはエラーを見つけるテストである）。
- 自動的にシステムのテストが行われるようにせよ。
- システムの性能に関して、開発に関与していない第三者専門家に妥当性を検査してもらえ。
- エキスパートシステムを、既存のシステムや自動化されてないメソッドと、ある一定期間平行して使用し、その結果を比べよ。
- システムがたとえその時点で機能の一部分しか提供していなくても妥当性検査を早く開始し、システムが完成するにつれて、回帰テストを継続せよ。
- システムの能力に関する詳細な情報をもつこと。

（６）まとめ

Ｖ＆Ｖの鍵は要求仕様をつくり込むことである。そしてプロトタイプはエキスパートシステムが直面しているぼんやりとした、明確に定義されていない問題について比較的特化された要求仕様書を得る方法である。

Ｖ＆Ｖは開発過程で実施される多くのステップと並行して行われ、繰り返し的に行われるプロトタイプ開発方法論にたやすくフィットする。

5. 専門家間の概念構造の比較

タイトル： Comparing the Conceptual Systems of Experts
著　者　： Gaines,B R and Shaw,M L.G
出　典　： IJCAI-89,pp 633-638,(1989)
分　類　： Credibility and Verification, Life Cycle Issues
ｷｰﾜｰﾄﾞ　： 専門家概念、実体・属性グリッド、ﾌﾞﾚｰﾝｽﾄｰﾐﾝｸﾞ方式

（１）概　要

知識ベースシステムの開発は複数の専門家から知識を獲得することが多い。この場合、専門家によって概念や語彙の使い方が一致していないために暗黙の混乱を引き起こすことがある。この論文ではこのような問題に対し、専門家間の不一致が簡潔に理解できる方法を提案し、複数の専門家からの知識獲得に対する実際的なテクニックを述べている。

（２）専門家概念の比較により生じる異論

専門家が持っている概念を実体-属性グリッドで表し、それを比較することによって知識を抽出することは、専門家の宣言的な知識構造を獲得するための強力な方法である。この実体-属性グリッドでは専門家から２つの型の情報を抽出している。一つは関連した実体間によって形成される特徴であり、もう一つはその特徴に対して使われる用語である。

ここで、専門家の概念から対象とする領域の関連した特徴への写像と、その特徴に対して使われる用語への写像とによって起こるいくつかの現象を考える。

①一致……専門家概念が同一の特徴に対し同一の用語を割り当てる場合
②衝突……専門家概念が異なる特徴に対し同一の用語を割り当てる場合
③照応……専門家概念が同一の特徴に対し異なる用語を割り当てる場合
④対照……専門家概念が異なる特徴に対し異なる用語を割り当てる場合

一致は専門家概念が共通していることを示している。衝突は対象とする領域において概念の適用が異なっているので、異論の可能性を示している。照応はある概念に使われる用語が異なっているので、これも異論の可能性を示している。対照は専門家概念が共通していない、すなわち専門領域が異なっていることを示している。したがって、直接的に異論を導くことはない。

（３）専門家概念の比較

専門家概念の構成要素となる各々の実体-属性グリッドは２極値て表されており、ファジィー集合のメンバーシップの度合いて定義されているものとみなす。例えば、ある実体に対応する"不連続－連続"という属性があるとすると、不連続から連続までの程度を５段階で表現している。

また、２つの属性間の同等の測定は次のように定義する。

$$\forall e\,(x \equiv y) = \bigwedge_{e}(x_e \equiv y_e) \qquad ①$$

本稿の例では同等に関して Lukasiewicz's operator を使用し、結合に関して Zadeh's sigma count を使用している。

単純化のために最初に２人の専門家の１対比較について述へる。（２）項で示した一致と衝突はある特徴に対して共通の用語を要求する。それは、実体－属性グリッドの交換に基づいて定められる。まず一人の専門家に対して別の専門家から抽出したグリッドを与える。そして、与えられた属性に対する実体の程度を求める。式①は２つの概念が照合することによって形成される特徴が良いかどうかの尺度として用いられる。しきい値は２つ設定してあり、上の方のしきい値以上に照合する特徴は一致を示している。下の方のしきい値以下に照合する特徴は衝突を示している。２つのしきい値間に照合する特徴はあいまいに関連していることになる。

照応と対照については２つの専門家概念間から独立しているグリットを抽出することによって測定している。式①を用いることにより、あるグリットが他と最も良く照合するものを各々の特徴に対して決定する。先程と同様に、しきい値を２つ設定する。上の方のしきい値以上に照合する特徴は照応を示している。下の方のしきい値以下に照合する特徴は対照を示している。２つのしきい値間に照合するときはあいまいに関連していることになる。

一致、照応、衝突、対照への誘導は、グリットにあるデータに基づいており、完全にアルゴリズミックである。この誘導はナレッジエンジニアではなくプログラムによって行われる。その根拠は視覚化された出力を通して専門家に明示される。ここで発生した異論については、各専門家が熟考し、お互いに議論することになる。そして、必要があれば変更したり保留したりする。

本文ではこの過程について地理のマッピングを例に具体的に述へてある。

（４）概念構造の比較方法

（３）項では２人の専門家での比較方法を示した。しかしｎ人にはｎの知識があり、１つの基本となる知識に対して n-1 回の交換が生しるので、グループの規模によって線形的に大きくなる。この項ではグループての知識獲得に対する実際的なテクニックを３つのフェーズに沿って述へている。

フェーズ１：領域の論議と具体化

①グループで対象とする問題を議論し問題領域を統一する。
②コンピュータ利用による知識獲得法を導入し、特定のタスクを定める。
③各々の専門家が実体-属性グリットの初期設定を行う。
④これらのグリットを（３）項て説明した方法て比較を行う。
⑤比較結果についてグループで討議し、抽出する知識の目的等の改訂を行う。

フェーズ２：概念化とフィードバック

①共通の実体集合を用いて、各専門家からグリットを抽出する。
②このグリットで再び比較を行い、その結果を各専門家に提示する。
③このデータは照応あるいは対昭の属性となる。

フェーズ３：交換と比較

①ある専門家概念が他の専門家概念の属性を用いて共通の実体集合を分類し、フェーズ２で抽出したグリットの交換を行う。
②このデータは一致あるいは衝突の属性となる。
③専門家概念と専門家概念間の関係をグループで議論する。一致については問題はない。衝突は分析され、解決できないものとして注を付ける場合もある。

照応は用語の違いによるものか、いくつかの現象を通しての相互関係によるものかとして解釈される。対照については、無関係、あるいは何人かの専門家の見地の関連したものとして解析される。

④この分析は、クラス、属性、オブジェクトの定義やルール記述といった知識獲得のさらなるフェーズを通して行われる。

（５）結　論

ここで提案した方法は複数の専門家概念の類似と差異を容易に表すことが可能である。また議論の焦点を合わすのにも使用することも可能である。グループで発生する異論は異なる用語や抽象レベル、不一致なもの等を分類出来るようにして問題解決を行なわなければならない。

グループによって貴重な知識体系を開発していくためには、各人の異なる見地を尊重し、ブレーンストーミング方式で専門家同士が影響を及ぼしあいながら専門領域の概念的体系の十分な探求を進めていくことが必要であろう。個別に抽出した専門家概念を客観的に分析することにより、社会的圧力による無意味な干渉を防ぐことにもなる。

6. 決定理論を用いた経験的知識の正当性の評価法

タイトル	Using Decision Theory to Justify Heuristics
著者	Langlotz,C P , Shortliffe,E H , and Fagan,L M
出典	AAAI-86,pp 215-219,(1986)
分類	Method and Techniques
ｷｰﾜｰﾄﾞ	決定木,感度解析,期待利得

（１）概　要

本稿では、経験的知識(Heuristics)を『状況(situation) →行動(action)』という形式で表現する場合の功罪(merit) を決定理論を用いて行う手法について述べる。具体的な説明をするために、ＭＹＣＩＮシステムで実際に使われているルールを基に、その意味表現(Context)を用いた解析を行う。

また、経験的知識に対する従来の議論とを対比させながら論しる。人工知能と決定理論の融合がエキスパートシステムにおける知識の評価基準を与え、その能力を高め、エキスパートンステムが適応できる問題解決領域(problem solving domain)を広げることになろう。

（２）はしめに

知識ベースにルールを登録する場合にルールの推論能力を確かめる方法やシステム性能に及ぼす影響度合いを解析するための標準的な手段がない。完全なブラインド評価の研究ができるまで、エキスパートンステム構築者はシステムが出くわすであろう全て（殆ど）のケースについて、その分野の専門家(domain expert) から集めた経験的知識が適切であるかどうかを検討しなければならない。

最善の行動(recommends an action)をとるための経験的知識の決定理由を作成するためには、その行動の影響に対する見込みと望みの両方を考えることが重要である。それゆえ、経験的知識の解析には人工知能と決定理論の融合が重要であると主張するゆえんである。決定理論は、確率(liklihood) と利得(desirability)の積として、次式のように表現される。

$$期待利得 = \Sigma_i P(O_i) \times U(O_i) \qquad 1)$$

ここで、$P(O_i)$ はプランを実行した時の i 番目の結果の起きる確率

$U(O_i)$ はプランを実行した時の i 番目の結果の利得

この期待利得を最大とする行動を選択することになる。

（３）経験的知識の形式的な解析

議論のために、ここではＭＹＣＩＮの経験的知識のルールを引用して使うことにする。

ＩＦ：　１）考慮中の治療法がテトラサイクリン(tetracycline)の投与で
　　２）患者の年令が８才以下

ＴＨＥＮ：テトラサイクリンによる生体組織に対する治療は、適切でないとの強い確証（０ ８）がある。

この経験的知識は幾つかのサポート知識（部分知識）の集合として表され、

そのサポート知識によって評価される。この場合は、４つのサポート知識として表現される。

若年者に対するテトラサイクリンの投与
＝＞　発達中の骨では薬のキレート環（化合物）ができる。
＝＞　歯の変色が起きる。
＝＞　体の変調を起こす。
＝＞　テトラサイクリンの投与は認められない。

最後の推論（結論）は、その前の推論に基づく推論としてなされる。これらのルールによる推論は、推論（ルール）の粒子の精度（知識の断片のサイズ）には係わらず、常に『状況→最善の行動』というチェーニングによって行われる。例えば、テトラサイクリンの投与を避けることが歯の変色の可能性を本質的に除外するのに有効であるが、治療効果の弱い他の薬剤を必要とすることになる。実際、テトラサイクリンは他の薬剤では効果が出ない場合や禁忌（治療法が望ましくない場合）の状態を除き、８才以下の子供には投与しないように医者の本には書かれている。言い換えれば、薬剤の投与は歯の変色というリスクと治療効果とのトレートオフの選択と言える。このトレートオフが薬剤投与をする場合の決定に対し非常に重要なことになる。

（４）問題の定式化

この項では、ＭＹＣＩＮのルールの例を基に、前項て述へたトレートオフについて決定理論を用いて解明する方法について述へる。決定理論での問題の定式化を行うためには、次の３つの質問に答える必要がある。

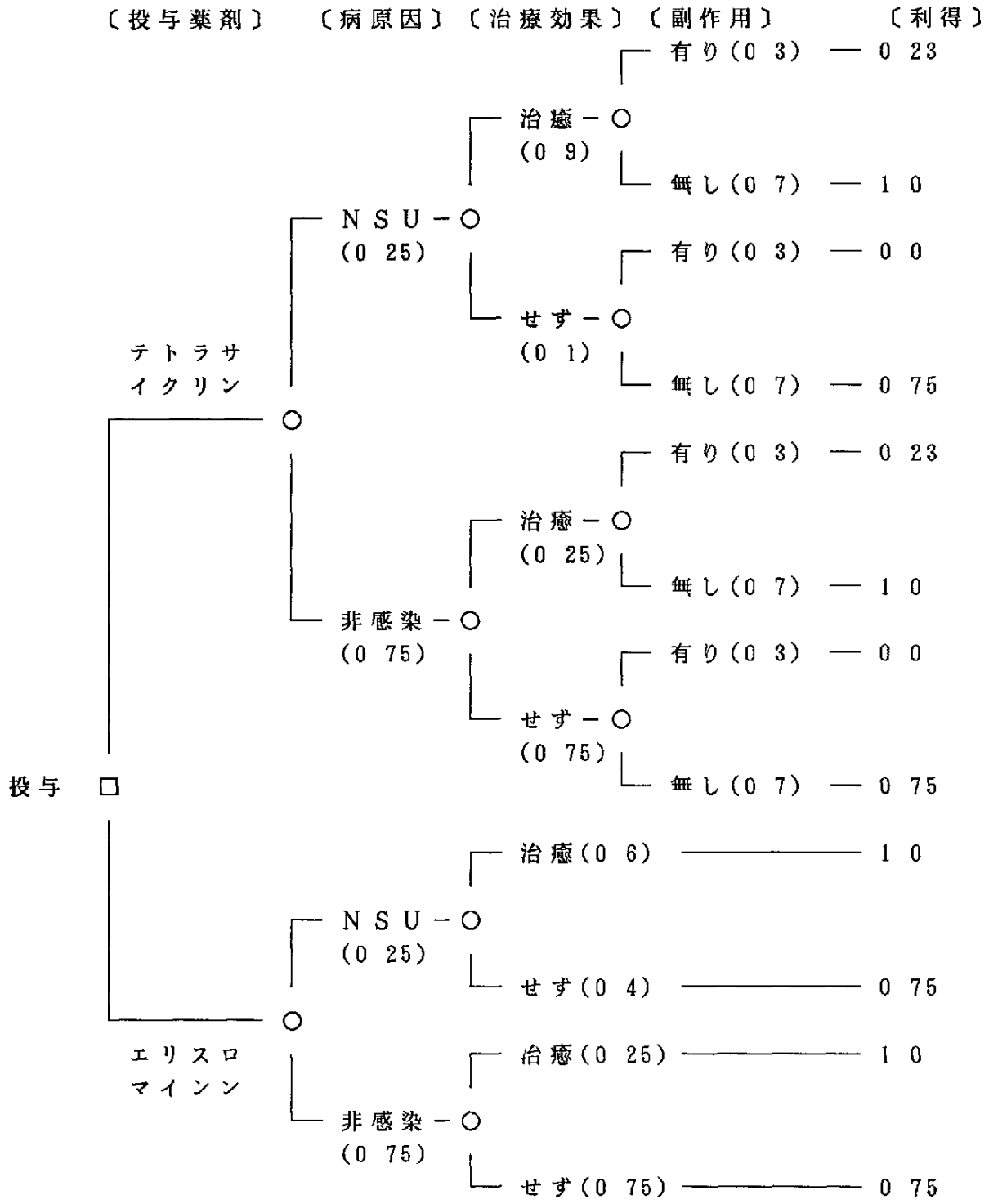
①どんな代替案の可能性があるか？
②それらの行動が実行された場合、なにが起きるか？
③それぞれの結果の利得はなんであるか？

我々は、テトラサイクリンのルールをもとに実際の例に応用して実験してみる。それによって、エキスパートシステムを構築する場合に実際に使用できるように、結果を如何に一般化できるか示してみる。

実際の例として、８才の男の子て尿道排泄（尿道感染の可能性もある）の患者を考える。しかし、培養試験によってバクテリア感染の確証はない。この場合、この尿道炎(Urethritis)は培養試験では簡単には分からない生体組織に起因するもの（ＮＳＵ：Non-Specific Urethritis)か非感染型の尿道の炎症が原因で起因するものかが考えられる。このような徴候に対し、若し大人の患者なら一般的に効果があるテトラサイクリンを投与すればよいので、悩む必要はない。しかし、この若い患者にはテトラサイクリンを投与するのか、他の薬剤を投与すへきなのかの決定を悩むことになる。ここにエリスロマイシン(Erythromycin)という薬剤がある。エリスロマイシンは、テトラサイクリンと違って、時々吐き気がするという副作用以外に歯が変色するような体にダメージを与える副作用はない。しかし、通常の尿道炎（ＮＳＵ）には殆と効果が無い。この問題の決定論的表現として定式化すると、薬剤投与の結果として患者は、次の四つの結果パターンが考えられる（stainingを副作用と訳す）。

①ＣＵＲＥ／ＮＯ　ＳＴＡＩＮＩＮＧ（治癒／副作用無し）
②ＮＯ　ＣＵＲＥ／ＳＴＡＩＮＩＮＧ（治癒せず／副作用有り）
③ＣＵＲＥ／ＳＴＡＩＮＩＮＧ（治癒／副作用有り）

④NO　CURE／NO　STAINING（治癒せず／副作用無し）
これらを基に、決定木(decisin tree)を書くと図1のようになる。



NSU Non-Specific Urethritics

図1　決　定　木

（５）確率と利得の査定

決定木における確率と利得の決定は、その結果が発生する確率とその結果となった場合の得られるであろう利益（利得）として決められる。この薬剤投与の場合は、薬剤投与によって副作用（変色）がなく治癒する結果(CURE/NO STAINING)の利得を１０、副作用だけが起きて治癒しない結果(NO CURE/STAINING)の利得を００とする。副作用（変色）がないかわりに治癒しない結果(NO CURE/NO STAINING) の利得と副作用（変色）が起きるが治癒する結果(CURE/STAINING) の利得は、００～１０の中間値となり、それぞれを選択する確率から、それぞれ0.75、０23と決められる。

確率は徴候から推定される原因（病名）とその病気に対する薬剤投与の有効性などから導かれる。この薬剤投与の場合は医者からそれらの情報を収集した。

テトラサイクリンとエリスロマイシンの投与に対する期待値は、式①を適用して以下のようになる。

テトラサイクリンの期待値　＝　０63

エリスロマイシンの期待値　＝　０83（0 15+0 075+0 1875+0 421875）

テトラサイクリンとエリスロマイシンの投与に対する期待値の差０2は何を意味するのであろうか。

（６）感度解析

ここで、考慮しなければ行けないことがある。この事例の場合でも、各々の施薬によって変色の確率が変化し、それに伴って利得も変わるであろう。実際には、閾値(threshold) によって利得が量子化されて使われている。歯の変色の確率と期待利得との関係を図２に、モンテカルロシミュレーションによるテトラサイクリンとエリスロマイシンの期待利得の差を図３に示す。

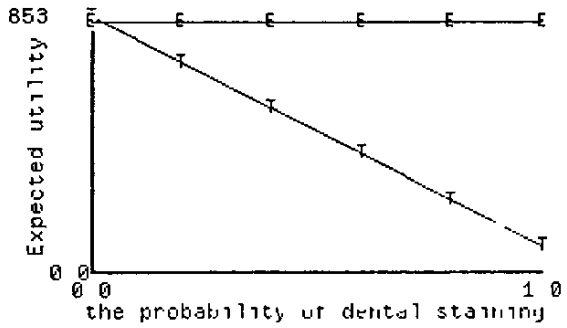


図２　感度解析の結果

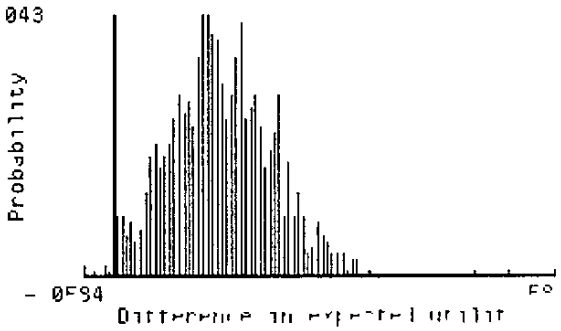


図３　薬剤による期待利得の差

（７）エキスパートシステムとの関連

細分化したルール(support rules) のシナリオだけでも幾つもあり、それらの評価が成されていないのに、ＭＹＣＩＮは何故うまくいったのだろうか？ＭＹＣＩＮのルールが選択のための制約となっており否定的に使われているのに対して、モンテカルロシミレーションの結果も、テトラサイクリンの投与の方が良い場合が非常に少ないことが裏付けられた結果となっている。また、実際の専門家との比較においても一致しているからである。

（８）まとめ

決定理論がエキスパートンステムの開発者にとって考えなければいけない非常に重要な道具であることを示してきた。決定理論が知識工学と結合することによって、決定理論を用いた手法は知識工学の基礎的役割を成すとともに、本質的なトレドオフ問題の解法や推論の説明昨日などシステムの能力を高める手法としても有効である。

経験的知識の解析に決定理論を応用する方法を提案して、実際の例でその様子を示してきた。この解析手法に必要な様々な情報は、従来のＫＥが行う経験的知識の通常の評価作業に示唆を与えることができるてあろう。

7. エキスパートシステムの妥当性試験手法

タイトル：	Validation of Expert Systems
著　者　：	O'Leary,D E
出　典　：	Decision Sciences,Vol 18,No.3,pp 468-486,1987
分　類　：	Verification, Validation
ｷｰﾜｰﾄﾞ　：	知識工学，エキスパートシステム，知識ベース，妥当性試験

（１）概　要

ＥＳ妥当性試験の１つの枠組みを提案している。この枠組みはＥＳの特徴から導きだしたもので、９つの分類からなる。

また、他の従来のＥＳ妥当性試験枠組みとの比較を行うとともに、ＥＳ妥当性試験に関する今後の課題も述べている。

なお、本論文は以下の構成である。

①ＥＳ妥当性試験の枠組みを明らかにするためにＥＳの特徴を利用する考え方を述べる。

②ＥＳ妥当性試験法の一般的な特徴を述へる。

③ＥＳ妥当性試験の枠組みを提案する。

④ＥＳ妥当性試験に関するコメントを述へる。

（２）はじめに

ＥＳ妥当性試験に必要な事項のうち、従来のソフトウェア試験と異なるのは以下の項目である。

- 何を知っているか、何を知らないか、何を間違って知っているかを確かめること。
- 専門知識のレベルを確かめること。
- 特定の領域での理論的な基盤を持っているかを確かめること。
- 信頼性を分析すること。

そして、その他の従来のソフトウェアと同様な項目は以下の通りてある。

- ユーザインタフェースの品質を確かめること。
- ドキュメントを妥当性試験すること。
- どんな記述言語、構築ソールが開発に適しているかを決めること。
- プログラミング（製造）の品質を分析すること。

（３）ＥＳ検査・試験の特徴

もし、ＥＳが従来の計算機ソフトウェアと同し特徴を持っているなら、従来のソフトウェアと同一の妥当性試験の方法を適用すへきだが、以下の３つの側面で異なった特徴があるため、これら特徴に着目して妥当性試験の枠組みを決める。

①技術的な側面

- 数値計算と異なった、記号処理にもとついている。
- 全知識の１割程度の知識が標準的な記述でなく、特殊な表現を必要とする（特別な機能を盛り込む）。知識の特殊な表現を適用しない場合には、

このような問題は生じない。

- ＡＩ言語やＥＳ構築ツールなどを用いてＥＳの開発を行う。

②環境的な側面

- ＥＳは、従来のように結論に間接的に寄与するわけてはなく、結論に直接影響を与える。
- 経験的な知識は直接的て、断片的である。

③適用領域に関する側面

- 科学技術計算と異なり単一の解のみを出力するわけてはない。
- 得られた結論の内容を評価する方法としては、他の問題解決手段とどのくらい結論が似かよっているかで行われ、結論の優先度を明確につける手段がない。
- 取り扱う分野で定説がないため、複数の専門家全員が合意する結論になりにくい。
- よい推論過程から、常に良い結論が得られるわけでなく、その時点で得られる最良の結論が得られるだけで、他の情報が得られた後ても有効な結論であることを保証するわけてはない。

（４）妥当性試験の枠組み

ＥＳ妥当性試験の過程は以下のような点て、従来のソフトウェア試験との違いがある。

①従来のソフトウェア試験では、プログラミングのエラーを見つけることが主体であったが、ＥＳの妥当性試験では何ができ、何がてきないかを確かめることになる。

②ＥＳではその記述言語・ツールが各種あり、それぞれについて少しつつ異なった試験方法が必要となる。また、試験方法はその実行環境にも依存する。

③ＥＳの開発手法が従来のトップダウンやボトムアップとは異なっている。

ここでは上記から、以下に示す９つの分類でＥＳ妥当性試験の枠組みを提案した。

1．知識ベースの内容の妥当性(content validity)　〔手段て分類した〕 a　専門家による知識ベースの内容の直接確認 b．人間の専門家との比較（チューリングテスト） (1)知識を提供した専門家との比較 (2)知識を提供した以外の第三者専門家との比較 c　他のモデルとの比較

〔方法〕

1．a.の具体的な方法としては、周期的なレビューや推論過程のまとめ、知識ベース内容の直接確認などがある。

〔課題〕

1．aの課題としては、専門家は計算機に慣れていない、人間の処理能力は大きくない、充分なリソースを必要とする、単純作業である、知識ベースが多いと多大な時間を要する、現状の技術てはてきない部分も多い、なとが

ある。

1．b.の課題としては、必ず個人のバイアスがかかる、定説の無い部分については第三者は確認が難しい、などがある。

1．c.の課題としては、もともとＥＳは同じ問題に対して他のモデルよりも同等か、より良い結果を得るため、あるいは、従来のソフトウェアでは記述しがたい部分に適用しているため、数値計算処理などでは実現が難しい、などがある。

2．基準に対する妥当性(criterion validity)
- a．専門知識のレベルの規定
  - (1)人間の専門家を基準とする
  - (2)レベルを判定するテストプログラムを利用する
  - (3)前もって定義しておいた応答の質を利用する
- b　知識ベースの基準
  - (1)一貫性
  - (2)正確性
  - (3)完全性
- c　妥当性試験の基準の分類

〔課題〕

専門家とＥＳとの専門度のレベルの規定に関しての違い、知識ベースへの基準の違い、どんな項目について妥当性試験すべきかが不明確、など各種の問題がある部分である。

2．a.の課題としては、専門家のだす結論が多くの条件の組み合わせからなるため、妥当性試験の基準と結論とを１対１に対応できない、応答の質を計ることが難しい、などがある。

2．c.の課題としては、一般的な妥当性試験基準を設定てきないことがある。

3．構成の妥当性(construct validity)〔どんな理論に基づいているか〕

〔課題〕

経験のみに依存する知識には適用できないことが課題である。すなわち、専門家が同じ理論にもとづいているか、という問題があるからである。

なお、一般的には、理論的な裏付けのないＥＳは失敗するといわれている。

4．客観性(objectivity)
- a　プログラマによる確認
- b　独立な管理組織よる確認
- c．開発要求元、エンドユーザによる確認
- d．バイアスとブラインド
- e．開発時に使用したテストデータとは異なるテストデータ

〔課題〕

4．aの課題としては、プログラマは問題領域知識を持てないことがある。このため、他の手法との組み合わせることが必要である。

4．bの課題としては、モデルの作成者による確認は不適当であることがある。

4．c.の課題としては、開発要求元、エントユーザが知識を有する専門家でない場合は不適当であることがある。

5．経済性(economics, cost-benifit)

ＥＳの導入効果や開発工数で定量化できるので有用である。ＥＳの妥当性試験のコストは、どれだけ厳密に試験するか、どの範囲まで試験するかで決まる。

〔課題〕

システムの導入効果は、そのシステムが使われるかに依存するので、特に実験的なプロトタイピングシステムではこの観点での妥当性試験は難しい。

6．信頼性(reliability, stability, ability) a．感度解析 b　標準的なテスト問題による確認

〔課題〕

6．aの課題としては、いろんな条件で多種の出力を得られるという現象が、ＥＳが良いためなのか、または、ＥＳが不安定なのか、との切り分けが難しいことである。

7．試験の誤差(systematic variance, experimental variance) a　取り扱う問題の範囲を反映したテスト問題 b　テスト問題の変化範囲 c　テスト問題の数 d　第Ⅰ種の誤りと第Ⅱ種の誤り

テスト問題の範囲は、ＥＳが実際に取り扱う範囲にする必要がある。

統計的な妥当性試験には、テスト問題は多数必要である。

8．その他の誤差(extraneous variance) a．ＥＳの複雑さ b　開発のライフサイクルでの位置（段階） c　知識の特殊な表現に依存する部分(fixes) d．試験判定者のいる場所 e．試験判定者が試験中に学習する効果

8．dとは、実験室で試験を実施をする場合、ＥＳを比較すへき試験の判定者の環境は簡単に調整できるが、現場で実施する場合はＥＳにとってより厳密な試験となるという違いがある。

8．e.とは、比較の対象となる試験の判定者が、試験に使用したテスト問題の操作やテスト問題のタイプから学習することである。学習により、試験の判定者がテスト問題の操作を変えて評価したりする。この問題は、テスト問題を経験した専門家と、ＥＳ操作の経験の少ない他の専門家との結論の差で決めることができる。

〔課題〕

8．aの課題としては、ＥＳの複雑度は規模で規定しにくいことである。

8．b.の課題としては、ＥＳ開発の初期段階では妥当性試験は重要ではないが、完成が近づくと妥当性試験は重要になってくる。すなわちどの段階でどの程度の妥当性試験をすへきか、などである。

8．cの課題としては、経験的には1割程度が標準的な知識の表現てないため、完全な知識ベースを実現するために特殊な表現が使用される（特殊な機能がンステムへ組み込まれる）。これらをまとめて試験するために、特別な試験方法を開発することになるが、これら自体が、もともと特殊なため、一般化できにくいことである。

9．エラーによる誤差(error variance)

「エラーによる誤差」については、各種の定義がありまた定着していない。

専門家のちょっとした思い違い、記憶違いなどで起こるもの全てを一緒にしたもので、ＥＳの妥当性試験の枠組みとして「その他の誤差」に分類できるよう「エラーによる誤差」を少なくしようと試みてはいるが、まだてきていない段階である。

（５）ＥＳ妥当性試験に関するいくつかのコメント

①蓄積済みの知識の内容の妥当性(content validity)

今後は知識ベース内の個別の知識に着目した場合にもコスト増加が少なく、また、他の専門家でもＥＳ全体としての性能を簡単に妥当性試験てきるようなツールの出現が望まれる。

ＥＳの妥当性試験は専門家に頼っており、また、ＥＳが拡張性を持つことから、少人数の専門家が担当している。さらに、知識を提供した以外の専門家による試験では、その専門家が該当対象に対して異なった経験知識をもっている可能性があるため、結論に意見の相違をもたらすおそれがある。

いくつかのケースでは、少人数による妥当性試験を、他のモデルや標準的なＥＳ妥当性試験法で補っている。しかし、他のモデルを開発することは人間の専門家による試験よりもコストがかさむ。

②基準に対する妥当性(criterion validity)

一般的には、ＥＳと人間の専門家からだされた結論には同一の妥当性試験基準が用いられている。この方法てはＥＳが人間の専門家よりもより良い結論を導くかもしれないという可能性は無視されている。通常、人間の専門家は全く別なモデルで業務を実施することは少ない。そこて、もしＥＳが人間

の専門家と全く同じように業務実施可能であるなら、妥当性試験基準を得る目的で別なＥＳを開発することもありうる。

プロトタイプシステムの開発者は妥当性試験そのもので忙しいので、妥当性試験基準を分類することは難しい。なお、現在までに第三者の立場の妥当性試験者により効果的な妥当性試験基準を開発したとの報告や、テスト問題の難しさのレベルを規定することが、ＥＳの問題解決能力を確認することであるとの報告もある。

しかし、ＥＳの問題解決能力を決定する一般的な方法はまだない。

現在までの報告では、妥当性試験基準としてＥＳのたす結果の正確さのみに着目していたが、完全性や一貫性に着目した報告もある。今後ツールはこれらの側面に焦点をあてて研究を進めるべきである。

③構成の妥当性(construct validity)

いくつかのＥＳで理論的な構成を試みてはいるものの、意思決定モデルの効用を無視して、どのように結論が導かれたを確認するだけてある。このアプローチの危険性はモデルが専門家固有のものであるため、一般化したり、他のＥＳや人間の専門家との比較を行っていないことにある。

ＥＳ開発に効果のある方法は、モデル化は済ませた結論を導く過程の性質をすることである。

④客観性(objectivity)

現在一般的には、妥当性試験は第三者的な独立の妥当性試験者を設置していない。このため、妥当性試験作業に当該ＥＳの開発担当者が支援をしており、バイアスがかかり、結論へ影響をおよぼしている。

⑤経済性(cost-benifit)

妥当性試験のための工数は定義てきるが、妥当性試験の効果をはかるのは難しい。

⑥信頼性(reliability)

ＥＳが開発段階であれば再妥当性試験のためのテスト問題を作成する必要はないが、実用システムでは知識ベースを変更後はテスト問題て再妥当性試験をする必要がある。しかし、知識が追加されているのて同じテスト問題でも別な結論を得ることがある。

⑦試験の誤差(systematic variance)

ＥＳの試験に用いられているテスト問題の数が少なく、また、典型的な例についてだけの試験である。これは、テスト問題のパラメータの変化できる範囲が限られているためで、現実に遭遇する極端な状況下や、あらゆる状況の変化の下でのＥＳの動作を試験しているわけではない。また、プロトタイプシステム段階での妥当性試験ては、内在している第Ｉ種の誤り、第Ⅱ種の誤りや、テスト問題の数を考慮していないことに問題がある。

（６）今後の課題

人間の専門家と同し結論を得るための妥当性試験基準は何か、ＥＳの専門度のレベルをどうやって規定する、が今後の課題てある。

8. エキスパートシステムの検証及び妥当性検査

タイトル： Verification and Validation of Expert Systems
著　者　： Green,C J.R. and Keyes,M M.
出　典　： WESTEX-87,pp 38-43,(June 1987)
分　類　： Method and Techniques, Verification, Validation
ｷｰﾜｰﾄﾞ　： 要求仕様書、エンジニアリング分析、相互作用分析、一貫性分析、不確実性分析

（１）概　要

検証及び妥当性検査（Ｖ＆Ｖ：Verification and Validation）は、コンピュータプログラムがユーザ要件を満たしているかどうかを決定するのに使用される形式的な手法である。エキスパートシステム（ＥＳ：Ｅxpert Ｓystem）の受け入れでの障害は、ＥＳ用のＶ＆Ｖ方法論の欠如である。ＥＳに対するＶ＆Ｖの実施は、「確固たる文書類の欠如」、「テスト結果を評価するのに使用する手法の不十分さ」、そして、「Ｖ＆Ｖ手法の開発を妨げる悪循環」のために、妨げられている。ＥＳに対するＶ＆Ｖを実施するのに受け入れられるＶ＆Ｖ手法があれば、ＥＳ技術の受け入れを助長するであろう。

ＥＳのＶ＆Ｖに関する提案された最初の手法は、「要件の定義」、「検証」、「テストケースの準備」、「テストの実施」、そして、「評価」からなる。

Ｖ＆Ｖの提案された方法論を洗練化し、顧客のＥＳに対する信頼を向上させるためには、ＥＳのサンプルに本方法論を適用する必要がある。ＥＳのＶ＆Ｖ実験の対象候補としては、顧客側でのＥＳに対する信頼度が増せば利益があり、そして、妥当性検査及び評価で採用されている戦術の問題点を追跡できるようなＥＳでなければならない。

（２）検証及び妥当性検査（Ｖ＆Ｖ）とは

検証及び妥当性検査（Ｖ＆Ｖ：Verification and Validation）は、下記の条項について決定を下すために行われるソフトウェアの公式のテストからなる。

①各レベルの仕様書及び出荷可能なプログラムコートが、より上位の仕様書へと追跡可能（traceable ）な状態にあること。即ち、仕様書あるいはプログラムコードが、より上位の仕様書の要件を完全に、かつ排他的に実現していることである。追跡可能性を決定する過程が検証と呼ばれる。

②出荷可能なプログラムコードが、本来のユーザ要件を正しく実現していること。出荷可能なプログラムコードがユーザ要件を満足している（あるいは満足していない）ことを直接に示す過程が妥当性検査と呼ばれる。

（３）エキスパートシステムのＶ＆Ｖを実施する際の問題点

Ｖ＆ＶをＥＳに適用する際の困難さのいくつかを、以下で吟味しょう。

①ＥＳに対するソフトウェア要求仕様書は、しばしば存在しないし、不明確で、素早く変更される。

②検証を成功させるには、より上位の仕様書の要件が、下位の仕様書の中に、少なくとも見つけられなければならない（そうでないならば、要件の追跡

はムダである。)。

③たとえ要件追跡が可能なほと十分に記述された仕様書が利用できたとしても、従来の検証では、構築されたシステムが要件をほんとうに満たしているかどうかを決めるに必要な多くの答を産出するかは疑問てある。

④従来の妥当性検査では、厳密なテスト手続きを要求している。

⑤ＥＳのテストの結果を評価する広く受け入れられた信頼できる手法がない。

ＥＳのＶ＆Ｖをどのように実施するかの理解が不足しているため、悪循環に陥っている。この悪循環を断ち、ＥＳ用のＶ＆Ｖ方法論を進歩させるには、Ｖ＆Ｖ技術を実験的にＥＳに適用することが必要である。

（４）エキスパートシステム用のＶ＆Ｖ手法

ＥＳ用のＶ＆Ｖに関する実務的かつ理論的な知識が不足しているので、適切な手法の選択として、実験的（cut-and-try）アプローチが最も有効である。本稿で提案する、ＥＳ用のＶ＆Ｖの第一次実験（first-cut）での手法は、下記の５つのタスクからなる。

①要件の定義

②知識ベース及び支援ソフトウェアの検証

③テストケースの準備

④テストの実施

⑤テスト結果の評価

１）要件の定義

ＥＳの開発は、しばしば、ソフトウェア要件に関する公式の定義なしで進行している。これは、下記の理由からてある。

①ユーザ及び開発者は、目的を記述した書類からプロトタイピングを始め、プロトタイピング過程で要件を洗練化することを望んでいる。

②ユーザあるいは開発者は、要求仕様書を準備したり、維持するのに必要な経営資源を積極的に提供するつもりがない。

③開発者は、強制力のある詳細に文書化された要件は、過度に、開発者の創造性を制約させると信じている。

④ユーザは、公式の受け入れテストを実施することを計画していない。

ＥＳ開発が有用な要求仕様書なしで進行している場合には、Ｖ＆Ｖ担当者は、作成されたソフトウェアから要件を規定することを試みる望ましくない立場にある。要件をソフトウェアに一致させるのは、政治的な圧迫があるかもしれないし、あるいは、ユーザまたは開発者は、要件を詳細に述へることに興味がないかもしれない。Ｖ＆Ｖ担当者は、開発者側の反対をおして、要件を規定するには、組織上の権限を必要とし、そして、ユーザ、開発者、及びＶ＆Ｖ担当者の３者の意見の一致を得るためには、この権限を行使しなければならない。

要求仕様書を関係者全員に受け入れてもらうためには、要件の作成、開発者側のレビュー、及び顧客側のレビューを何回もくり返す必要がある。有用な要求仕様書が手元にあるようになれば、Ｖ＆Ｖの残りのタスクを始めることが可能となる。

２）検証

従来のソフトウェアの検証は、要求仕様書、設計書、及びコートの階層を通過する要求条項の追跡と、設計の適切性を実証するエンジニアリング分析（設計解析）からなる。要求条項（要件）の追跡では、システム／ソフトウェア要求仕様書、設計書、そして、ソースコートが対象となる。正規に実施されるエンジニアリング分析では、致命度（criticality）、感度／安定度（sensitivity and stability）、効率性（efficiency）、そして、保守性（maintainability）が対象となる。ここでは、これらの活動が、ＥＳの検証の際にとのようにして実施されるかについて検討する。

①要求条項の追跡

要求条項の追跡は、上位の仕様書の要求条項のすへてが、下位の仕様書て満たさてい るかどうかと、下位の仕様書に余分な要求条項が入っていないかとうかを決定するタスクである。要求条項の追跡は、誤り、脱落、及び余分な機能が、ある開発工程から次の開発工程へと流れ出ていかないことを保証するために採用されている。

②エンジニアリング分析

【共通のエンジニアリング分析】

エンジニアリング分析は、しばしば、検証の一部として実施される。これらの分析は、コーディング作業やテスト作業を実施する前に、システムが意図した通りに動くかを予測するために採用されている。数種のエンジニアリング分析が、下記に示すように、特定のプロジェクトのニーズに応じて、実施される。

［致命度］このソフトウェアの障害が、他の構成要素あるいは使命達成に対し、どの位の影響を与えるか。

［感度／安定度］入力のほんの少しの変化、特に特異点や運用限界点付近ての変化に対し、採用しているアルゴリスムや試行錯誤手法がどの位敏感であるか。入力の少しの変化に対し、不正確、不安定、あるいは、失敗してしまうのか。数値解析ては、しばしば、アルゴリズムの感度の判定を行っている。

［効率性］ソフトウェアは、与えられた制限時間内で、かつ制限資源内で、意図された仕事を遂行することができるか。

［保守性］ソフトウェアは、そのライフサイクル全体を通して、保守を容易にするように作成されているか。

【固有のエンジニアリング分析】

従来ソフトウェアでは採用されていない３種類のエンジニアリング分析は、ＥＳの検証では重要となる。これらは、相互作用分析（ＩＡ：Interaction Analysis）、一貫性分析（ＴＡ：Truth Analysis）、そして、不確実性分析（ＵＡ：Uncertainty Analysis）である。ＩＡは、望ましい結果を生成するためのＥＳの構成要素間での相互作用に関する能力を取り扱う。ＴＡは、知識ベースの中味の一貫性（truth）を評価する。ＵＡは、ＥＳが不確実性の尺度を使用している際に適用可能である。この分析は、割り当てられた不確実値と、その不確実値を組み合わせる手法とが正しい結果を与えるかどうかを決定する。

３）テストケースの準備

ＥＳ用のテストケースの選定は、原理的には、従来のコンピュータプログラムと同じであり、かつ、困難さも同程度てある。Myersは、従来ソフトウェアのテ

ストケース設計に関して、詳細に述へている。ＥＳ用のテストケースの選定基準は、下記のように要約できるであろう。

①すへての要件をテストすること。

②ンステムの限界点の１つにンステムの状態を陥いらせたり、あるいは、誤り回復や後始末のルーチンを呼び出したりするような一連のテストケースは、独立のテストケースの集りよりも、誤りを指摘するのには、しばしば、より有効である。

③コートのすへての項目及びすへての可能な分岐をテストすること。

④ＥＳの場合には、ンステムの出力結果と、システムがその出力結果に到達した道筋の両方をテストする必要がある。

⑤すへての特異点及び境界条件をテストすること。

⑥条件の組合せに関して分別のある選定をしてテストすること。

⑦使命達成が第一でるようなシステム（ＭＣＣＰ）の場合とか、テスト予算が許容される場合には、ンステムが機能低下したり、あるいは、失敗したりする個所を同定するために、要求限界や設計限界を越えてテストすることも重要である。

４）テストの実施

ＥＳのテストの実施は、従来ソフトウェアのそれと同しである。オペレータは、テストケース準備工程で用意された台本（script）に則ってテストを実施し、そして、台本から逸脱したものすへてを説明を付けて記録する。テスト入力を用意し、テスト結果を自動的に記録する“台本”及び“ロク”ファイルを使用して、従来ソフトウェアのテストを実施するのが有利であることが証明されている。ＥＳを台本に基づいてテストした有望な結果が既に得られている。

５）評　価

ＥＳのＶ＆Ｖに関する論争の大半は、テスト工程ての評価段階に起因している。ＥＳの評価に関する問題点のいくつかを、下記に列挙する。

①ＥＳのテスト工程では、従来のソフトウェアのテスト工程て採用されている客観的で正確な期待される解及び評価基準が適用てきないかもしれない。

②受け入れ可能な解が何であるかの一致がない場合もある。

③特に、その問題領域で不確実性を扱う推論を含んている場合には、ＥＳは100 %の正解率て解を与えることはできないかもしれない。

独立の評価の必要性がはっきりした。ＥＳの開発に参画した同し専門家が、そのＥＳの評価に採用された場合には、第三者には、その評価結果ては満足てきないし、心理的なバイアスがないとはいえない。

ＥＳの評価で考慮すへき品質特性と、従来ソフトウェアて考慮すへきそれとは、かなり異なっている。ＥＳの評価て考慮すへき点を下記に示す。

①期待される解、等価な解、あるいは、受け入れ可能な等しい解が得られるか。

②ＥＳはその解に対して正当性を説明することを期待されている場合には、提出された正当化理由が、顧客が要求している質や様式に合致しているか。

③ＥＳに対する顧客の信頼を保証する観点から見て、ＥＳが辿る推論の過程が十分に完備的（complete）て、かつ頑健（robust）てあるか、そして、

採用されている知識が十分に深いか。

④ＥＳが100 ％の正解率を達成することが期待できないのならば、不正確な解に対するインパクトとは何なのか：不正確な解は単に準最適解にすぎないのか、ＥＳを無能にしてしまうほど悪いのか、あるいは、ＥＳが支援している使命達成を実際に損なってしまうのか。もし、不正確な解が許容される場合には、不正確な解集合のどの位の部分が受け入れ可能なのか、そして、不正確な解を受け入れた時の最悪の受け入れ可能な結末（consequence）とは何なのか。

（５）実験的Ｖ＆Ｖ

ＥＳ用のＶ＆Ｖの方法論の開発には、実験を必要とする。実験は、ＥＳを検証及びテストする際に採用する技法の洗練化を可能にし、その有効性面ての信頼を向上させる。

実験的なＶ＆Ｖ方法論が下記の能力を実証した暁には、ＥＳに対して実験的なＶ＆Ｖから本番用のＶ＆Ｖへと移行するものである。

①ＥＳのすべての開発工程でうめ込まれた誤りを検出すること。

②Ｖ＆Ｖをパスしたことにより、ＥＳの正当性を保証すること。

③ＥＳソフトウェアのライフサイクルコストの低減に貢献すること。

ＥＳを検証し、妥当性検査を行う能力は、ＥＳ技術の受け入れに十分に貢献する。

9. 知識ベースのテスト

タイトル：	Testing Your Knowledge Base
著　者　：	Marcot, B
出　典　：	AI Expert,Vol 2,No 8,pp.42-47,(July 1987)
分　類　：	Verification, Method and Techniques,Life Cycle Issues
ｷｰﾜｰﾄﾞ　：	ﾗｲﾌｻｲｸﾙ、評価基準（頑健性、予測的妥当性など）

（１）はじめに

- 本稿では、知識ベースのテスト方法と、その際に使用する妥当性評価基準について述べる。
- 知識表現としては、プロダクションルールのみを考える。

（２）テストの必要性

- テストは、プロダクションルール開発での必須の作業で、開発サイクルに組み入れるべきである。
- 検証（Verification）には、コンピュータコート（Lispコード、ルールベース）がバグなしで作成されていることを保証することが含まれる。
- 妥当性検査（Validation）には、ルールの意味や内容が充足性（Adequacy）に関するいくつかの慎重に定義された評価基準を満たしていることを保証することが含まれる。
- 知識ベースの開発サイクルは、下記の工程に分かれる。
  ①プロトタイプの創成
  ②第１世代ルールセットの開発
  ③テストと、ルールセットの第２世代への拡張
  ④第２世代ルールセットのテスト

（３）プロトタイプの創成

- プロトタイプは、典型的には、１ダース程度のルールから成り、その問題に対する知識工学的アプローチの実現可能性を実証する。
- テストでは、そのプロトタイプを評価して、この開発プロジェクトを続行させるべきか否かを決定する。

（４）第１世代ルールセット

- 第１世代ルールセットは、典型的には、数ダースから数100個程度のルールから成る。
- アルファテスト（Alpha-testing）の段階では、システムの正確性（Accuracy）と充足性（Adequacy）がどのレベルに達しているかを決定する。

（５）ルールセットの拡張

- 先ず、この段階では、フルスケールシステムに対する能力基準を決めること。
- 拡張作業には、典型的には、推論構造、高位の制御ルール、そして、低位の事実及びそれらの関係の再構築も含まれる。

（６）第２世代のルールセット

- 最初のフルスケール知識ベースに対するテストは、利用現場で行われる。
- ベータテスト（Beta-testing）の段階では、システムの正確性、精度（Precision）、及び信頼性（Reliability）がどのレベルに達しているかを決定する。

（７）市　場　性

- ここで言っている市場性（Marketability）は、特に、想定利用者へのシステムのアイデアや使用に関する売り込みについてのことてある。
- 妥当性テストの主要ステップには、下記の作業項目が含まれる。
  ①エキスパートシステムが使用される領域及び文脈の定義
  ②妥当性に関する個別の能力評価基準の同定
  ③妥当性検査の実施と、テスト結果の分析・評価

（８）領域・文脈

- 知識ベースがうまく解くことが期待される領域及び文脈を決定すること。
- モデル化においては、モデルが使用される諸条件を厳密に定義すること。

（９）個別の能力評価基準

- 能力評価基準は、知識ベース開発の各段階ごとに規定されること。
- テストを実施する前に、各能力基準の合格レベルを決定しておくこと。

（１０）テスト結果の分析

- 定量的テストを行う場合の分析段階には、問題領域、既知のテストケース、システムの実行方法、出力結果の記録方法などについての決定が含まれている。
- 妥当性テストの結果は、システム開発の評価にも当然使用される。
- システムが本来の目的や能力基準を満たしているかである。

（１１）妥当性検査の評価基準

- 下記に示す評価基準が、知識ベースの妥当性をテストし、評価する目的で使用されるかもしれない。

＊正確性（Accuracy）

- シミュレーションがどの位良く現実を反映しているかを表す度合てある。
- ルールを使用して推論された結果と、史実（既知）データとを比較し、結果の正当性を評価すること。

＊適応性（Adaptability）

- 将来の開発や適用がどの位容易であるかの可能性を表す度合てある。
- 新しい情報が利用可能になった時には、Ｉ／Ｏや制御ルールの汎用性を保持して、事実やルールを改訂すること。
- 既存の、あるいは提案されたハートウェアやソフトウェアンステムとの統合の望ましさを、定期的にレビューすること。
- システムが自己修正的であるのか、あるいは、文脈に敏感であるのかの、いずれなのか。
- システムが、特定のユーザニーズを満たすためにカストマイスてきるのか。

＊充足性（Adequacy）

- 関係する経験的知識の内て、どの位シミュレーションできるかを表す度合である。
- 推論結果に影響を与えるパラメータ群（変数、条件、関係（変数間の論理的関係及び統計的関係））のリストを設定し、ルールセット中にとのパラメータをいれるかを決定すること。

＊アピール性（Appeal）

- 使用性（Usability）のこと。
- 知識ベースが我々の直感とどの位一致し、我々の思考をとの位刺激するかを表す度合である。
- アピール性は、市場性（Marketability）に対してきわめて重大な影響を与える評価基準である。
- 開発工程の比較的早い時期に、Ｉ／Ｏの友好性（Friendliness）を評価し、使用性をテストすること。
- ベータテスト（Beta-testing）を行うへき開発段階において、現場て、シミュレーション及び実用性をテストすること。

＊入手性（Availability）

- 現実的な市場性を決定付ける際に重大な影響を及ぼす、同し問題を解決する、他のより簡単で、妥当な知識ベースがどの位存在するかを表す度合である。
- 他のツールが既に利用可能で、かつ、そのツールがユーザニースを満たしている場合、ユーザが新しいルールベースに対するニーズを感しているか。

＊規模性（Breadth）

- 知識ベースで使用されるルールの個数に比例する度合てある。
- システムが解くへきてあると期待される文脈の個数と、ルールセット中で考慮すへき関連のあるパラメータの個数を決定すること。

＊深　度（Depth）

- モデルの各構成要素を表現するために選択された変数の個数及び種類に比例する度合である。
- システムが注目する条件群の各々の範囲と、各条件下て、診断したり、分類したり、助言したりする際にどのパラメータが必要かを決定すること。
- 深度を決めたならば、そのたびに、必要な入力データ及ひユーザインタフェースを決定すること。

＊表面的妥当性（Face validity）

- モデルの信用性（Model credibility）のこと。
- プロトタイプ開発の初期の段階及びフルスケールシステムの拡張の最終の段階で、信用のある専門家が、知識ベース、推論構造、そして出力結果をレビューすること。
- そのレビュー結果は編集され、報告されること。

＊汎用性（Generality）

- 知識ベースが広い範囲の同種の問題を解決するのにとの程度利用てきるかの能力を表す度合である。
- システムが専門家レベルの能力で解くことができると期待される汎用的な文脈群を定義し、これらの文脈を越えて利用した場合には、正確な結果が与えられないかもしれない点について強力な警告が用意され、表示されること。

＊精　度（Precision）

- モデルが特定の諸システムパラメータをどの程度忠実にまねているかのモデルの能力を表す度合である。
- 数値変数や計算の場合には、有効桁数のことである。
- すべてのパラメータ群の関係する変動範囲すへてがルールへースや事実群でカバーされていることを保証すること。
- 数値は、必要があれば、浮動小数点や実数形式で表現されること。計算、特に、行列計算や線形代数計算の場合には、倍精度が用いられていること。

＊現実性（Realism）

- 関係する変数群や関係（変数間の論理的関係及び統計的関係）群をどの位考慮しているかを表す度合である。
- ルールベース中に入れるべきパラメータ群（変数、条件、関係）や機能群を、専門家やエントユーザが使っている用語や概念モデルを使用して設定すること。
- 現実性は、フルスケールの知識ベースを開発する際には、特に重要である。
- 現実性には、質問を出す論理的な順序も含まれる。

＊信頼性（Reliability）

- モデルによる予測結果が経験的に正しい割合を表す度合てある（実際は、ルールベース全体の正確性及び正当性は、高度な統計的分析て調へられ、信頼性はその一部である。）。
- 信頼性には、正しい診断結果及び分類結果に対する条件つき確率及び事後確率が含まれる。
- 最終的には、ルール及び出力結果に対する尤度の統計的効用を述へること。

＊分解能（Resolution）

- モデルが模擬を試みているシステムパラメータの個数に比例する度合である。
- 定義し、詳細に表現する必要のあるパラメータを、そしてより一般的な諸条件でくくることのできるパラメータを、あるいは無視できるパラメータを同定すること。

＊頑健性（Robustness）

- モデルの構造（あるいは外れ値の混入）に対し、特に敏感でない結論のこと（モデル化の際に決めた仮定がくずれても、能力が損なわれない割合を表す度合である）。
- どの入力パラメータが中間結果（診断、分類の場合）、最終結果（助言、警告の場合）、そして出力に対して、最小の、あるいは最大の影響を与えるか

を決定すること。
- 最大の影響を与える入力パラメータは、ルールベースや機能中で十分に定義されていることを保証すること。

＊感　度（Sensitivity）
- 知識ベースのパラメータ群の変動範囲内で、どの程度史実データと一致する出力を導出できるかを表す度合である。
- 特に、ある特定のパラメータを段階的に変化させ、その他のパラメータを一定に保持し、そして、モデルの出力を史実（既知）データに一致させるようにして、各入力パラメータに対する結果の感度を決定すること。

＊技術的及び操作的妥当性（Technical and operational validity）
- モデル化の際に決めた仮定に対して、認識された現実との相違を同定し、その相違がどの程度重大であるかを決めること。
- ルールや関係が前提とする文脈、条件、仮定を注意深く解明すること。
- 各々の仮定がモデルの結果をどの位制約するかを検討すること。
- 仮定がモデルの正確性、信頼性、頑健性、及び汎用性にどの位影響を与えるか。

＊チューリングテスト（Turing test）
- 特定の同し問題に対して、モデルによる結論と、（人間の）専門家が解いた結論とを、（人間の）評価者に区別してもらって、知識ベースの妥当性を評価すること。

＊有用性（Usefulness）
- システムが、問題解決の文脈下で使用するのに、必要十分なパラメータ群及び関係群を含んでいるかを調べること（少なくとも、モデルによる予測結果が経験的に正しいことがある場合）。
- 有用性に関しては、フルスケールンステムの場合には取るにたらないことではあるが、しかし、プロトタイピングや既存のルールセットへのルールの追加の場合には、重要てある。

＊予測的妥当性（Validity i e　Predictive validity）
- 知識ベースが、過去の経験に照らして見て、どの程度正しい予測結果を出すかを表す度合である。
- ンステムがうまく解くことが期待されている文脈が与えられた場合、との位多くの実際の条件で、システムが正確に診断したり、分類したり、助言したりできるかを決定すること。
- 同し対象領域で、（人間の）専門家が正当性のどのレベルを達成しているかを決定し、知識ベースの正当性に関して現実的な目標を設定すること。

＊総体性（Wholeness）
- モデルに組み込まれている（思考・推論）プロセス及び対話の個数に比例する度合である。
- ルールベースがどの位複雑か。

- それは、いくつの因子を使用しているか。
- 適応性をかんがみて、総体性を考慮すること。

10. エキスパートシステムの性能に対する妥当性検査

タイトル：	Validating Expert System Performance
著　者　：	O'Keefe,R M　Balci,O , and Smith,E P
出　典　：	IEEE EXPERT,Vol 2,No.4,pp.81-90,(Winter 1987)
分　類　：	Statistical Tools, Method and Techniques
ｷｰﾜｰﾄﾞ　：	定性的方法、定量的方法、多重応答問題（Multiple-response problem）、許容性能範囲（Acceptable performance range）、第Ⅰ種／第Ⅱ種の誤り

（１）概　要

エキスパートシステムの研究開発における要件として妥当性検査の確立をはかる。妥当性検査における問題点を検討し、定性的、定量的な方法を提示する。

（２）検証、妥当性検査、および評価

本論文では評価の一部である妥当性検査を取り上げる。たたし、全体的な評価と明確に分離することは困難な場合がある。また、検証ではシステムが仕様どおりに実現されているかどうかを実証するのに対し、妥当性検査てはシステムが許容できる程度の精度を発揮するかどうかを実証する。

（３）エキスパートシステムの妥当性検査

妥当性検査においては次の７項目の主要な問題点がある。

①何を検査するか（What to validate）

中間結果、結論、推論、およびこれらの組合せについて検査てきる。開発の初期の段階では推論の検査に重点がおかれ、知識ベースがより完全になったところで結論の検査を行うのが普通であろう。また、結論を正しいか誤りかのとちらかに分類するのは困難なことがある。このような場合でもHickamらのように数個のカテゴリイを用意すれば分類できることが多い。

②何との比較で検査するか（What to validate against）

既知の事実との比較て検査するへきてはなく、専門家の能力との比較て検査するへきである。

③何で検査するか（What to validate with）

テストケースの選択は検査に影響を及ぼす。生起頻度や困難さなとを考慮して包括的なテストケースを用意しなければならない。また、システム開発時に使われたものは検査に用いてはならない。

④いつ検査するか（When to validate）

エキスパートシステムは、開発の早い段階で許容性能を発揮すへきである。ただし、許容レベルは実験型プロトタイプと発展型プロトタイプとては異なる。実験型では基本的なアプローチの正しさが問題となる。発展型ては改良を必要とするところの洗い出しが重要てあり、応用がヘヒーデューティーてない場合には、フィールトテストによる検査も行われる。

⑤検査費用のコントロール（How to control the cost of validation）

検査には時間と費用とがかかりそれを予測することは困難てある。しかし、

フォーマルな妥当性検査方法を設計し開発プロセスと統合化することにより費用をコントロールできると思われる。

⑥バイアスのコントロール（How to control bias）

専門家の知識のバイアスを制御するには、ブラインド検査やクロスチェックを利用する。これらはOncocinの検査で用いられた方法である。システム開発者のバイアスを最小にとどめるためには、開発者とは別のチームによる検査を行う。

⑦複数の結果の取扱い（How to cope with multiple results）

複数の結果を出力するシステムを検査する場合には「多重応答問題」（multiple-response problem）とよばれる問題がある。例えば医療診断において２種類の薬品を処方した場合に、それぞれは妥当であっても併用してはならない場合などが相当する。

（４）基本概念

妥当性検査には次の５種類の基本概念がある。

①許容性能範囲（Acceptable performance range）

利用者が受け入れられる性能レベルは「許容性能」と呼ばれ、開発中に定めておく。システムがモデル化している専門家の能力に対応づけて決められることが多い。

②入力領域（The input domain）

開発者は特定の応用のためにシステムを設計すべきである。検査においてはその応用の見地から、定められた入力領域についての性能だけを検査すべきである。

③妥当性検査（Validity testing）

システムの妥当性は、入力領域内で、設計目的に関連して、許容性能範囲についてのみテストすることができる。

④検査のフォーマル性（Validation formality）

フォーマルな検査では、開発サイクルの中のどこで実施されるかが確立されており、また、検査方法、入力領域、許容レベル、および（適切な場合には）統計的手法の適用が定められている。

⑤開発者／利用者のリスク（Builder's/user's risk）

検査結果には第Ｉ種と第Ⅱ種の誤りが含まれる。第Ｉ種はシステムが妥当であっても棄却してしまう、いわば「開発者のリスク」である。第Ⅱ種はシステムが妥当でないのに受け入れてしまう「利用者のリスク」である。妥当性検査の設計にあたってはこれらについて客観的に考慮すべきである。

（５）定性的検査

定性的検査は主観的な性能比較を扱う。ここでは７種類のアプローチを示す。

①表面的妥当性検査（Face validation）

予備段階の検査法として有効な方法である。プロジェクトチームのメンバーがシステムの性能と専門家とを主観的に比較する。そのためにテストケースの結果と許容性能レベルとを表面的(at face value)に評価する。

②予測的妥当性検査（Predictive validation）

過去の事例を入力し、既知の結果あるいは専門家の判断と比較する。

③チューリングテスト（Turing test）

検査対象が専門家かシステムかをふせておいて、両者の比較で検査する。コンピュータを使っていることによるバイアスを除けるという利点もある。

④フィールドテスト（Field test）

プロトタイプを現場において動作させ、エラーを収集する。開発者側の利点は、利用者にテストの責任を負わせられること、および、許容性能範囲が暗黙のうちに得られることである。

⑤サブシステム単位での妥当性検査（Subsystem validation）

サブシステムに分割して検査できる場合には、開発と並行して検査が可能、検査の複雑さの減少、エラーの同定が容易、といった利点がある。

⑥感度解析（Sensitivity analysis）

入力やパラメタを系統的に変えることによって結果に及ぼす影響を調べる。感度解析は妥当性検査に用いられた実績がないけれども、強力な定性的評価法となり得る。特にテストケースが得られない場合や不確実性を導入している場合などに有効と思われる。

⑦視覚的対話（Visual interaction）

システムの動作を動画として表示し、対話的なパラメタの変更などを可能にすることによって、これまでに述べた表面的妥当性検査などの環境を提供する。専門家が推論プロセスを検査する際などに有効である。

（６）定量的検査

定量的検査では、統計手法を用いてシステムの性能とテストケースあるいは専門家との比較を行う。

①ｔ検定（２標本問題）（Paired t-test）

ｎ個のテストケースについて、平均値の誤差、標準偏差、および自由度ｎのｔ分布の値から信頼区間(confidence interval)を計算し、その区間にゼロが含まれていれば合格とする。

②ホテリングのT^2検定（Hotelling's one-sample T^2 test）

システムが出力する複数の結果の間に関連がある場合には、ｔ検定（２標本問題）では不適当でありホテリングの検定を用いるとよい。

③同時信頼区間（Simultaneous confidence intervals）

多変量応答システムの検査には、一対応答の違いに対する同時信頼区間を構成し、許容性能範囲との比較を行う。

④一致係数（Consistency measure）

複数の専門家との比較を行う場合には、専門家との間の一致性が問題になる。そのため、級内相関係数(interclass correlation coefficient)を利用して信頼性を評価する。

（７）まとめ

エキスパートシステムの開発者にとって、妥当性検査と評価とは現在よりもいっそう留意する必要がある。以下に有用なガイトラインの要約を示す。

- 許容性能範囲と規定された入力範囲とについて検査すること。
- 開発サイクルへ妥当性検査を組み込むこと。
- 利用者のリスクと開発者のリスクとを考慮すること。
- 適切な定性的検査を選択すること。
- 可能であれば定量的検査を用いること。

11. KB-Reducer: 知識ベースの非一貫性及び冗長性の検査に対する新アプローチ

タイトル：	Knowledge-Base Reduction A New Approarch to Checking Knowledge Bases for Inconsistency & Redundancy
著　者　：	Ginsberg,A
出　典　：	AAAI-88,pp 585-589,(1988)
分　類　：	Automation
ｷｰﾜｰﾄﾞ　：	冗長性の検査、非一貫性の検査

（１）はじめに

- 一人の専門家によて作られたエキスパートシステムが色々なサイトに配られ利用されることを考えた時、各サイトでそのサイトに合ったような知識の修正が行われることか考えられ、修正した知識による不具合が発生する可能性がある。
- KB-ReducerはＥＳ開発者に対し知識の冗長性、非一貫性を示してくれるツールである。

（２）KB-Reducerの推論モデル

- 知識ベースの一貫性、非冗長性を議論する場合は演繹推論機構が前提である。
- ここで述へる"KB-Reducer"の特徴は次の３つである。
  - ①単調性：ある入力データ集合αがＣを導くならば、αのスーパセットもＣ導く。ワーキングメモリ（ｗｍ）はassertのみでretractは無い。
  - ②無選択性：競合解消は無い。マッチしたものは全て発火される。
  - ③データ駆動型エンジンである。

（３）非一貫性及び冗長性の定義

- 論理で使われる一貫性の概念と知識ベースで使われる一貫性の概念の違いを抽出しておくことは重要である。

  論理に於いては、

  ｐ－> Ｑ

  ｐ＆ｒ－> ¬Ｑ

  は一貫性がある。

  しかし、知識ベースては一貫性が無いとされる。ｐとｒがインプットとして与えられるとＱと¬Ｑがｗｍにassertされてしまう。
- 一貫性に関しては知識ベースの方が厳格である。
- 冗長性に関しては論理の世界と知識ベースの世界とで同し様な概念てつかわれている。

①妥当な入力セットと意味的制約

- Ｅが知識ベースに入力されると意味的制約を犯していないかを検証する。

  ［例］John is a male human being

  John is pregnant ……男は妊娠しないと言う意味的制約を犯している

  オブジェクトの一属性は一つの属性値しか持たない(single-valuedness)

制約には論理的制約と数学的制約がある。

- トメインに固有の意味的制約知識は時間の関数として増えていく。従って今まで一貫性が検証されていても、意味的制約が追加されることにより一貫性が保てなくなることがある。
- 知識ベースが決して矛盾ある結論をassertしないためには意味的制約をベースに全ての潜在的に一貫性が保てないものを削除することが出来なければならない。

（４）KB-Reducer

- KB-Reducerはruleとfindingとhypothesesとdefault-hypothesesとから成っている。
  - rule
    ruleは右辺と左辺からなり、左辺ではand,orの結合子が使える。
  - finding
    findingは左辺にててくる文字及び右辺にててくる文字の論理否定てないものである。
  - hypotheses
    hypothesesは右辺にでてくる文字、または、いくつかのルールの右辺にあり且つ他のルールの左辺にある文字である。
  - default-hypotheses
    default-hypothesesは左辺にだけでてくる文字であり、且つあるhypothesisの論理否定であるものである。

(Knowledge Base)

$p \vee q \rightarrow R$

$a \rightarrow B$

$B \& R \rightarrow S$

$\neg S \& c \rightarrow \neg R$

Finding	p, q, a, c
Hypotheses.	$R, \neg R, B, S, T$
Default-Hypotheses	$\neg S$
Label for R.	$p \vee q$
Label for B	a
Label for S	$a \& p \vee a \& q$
Label for ¬S	$\neg a \vee (\neg p \& \neg q)$
Label for ¬R	$\neg a \& c \vee (\neg p \& \neg q \& c)$

図１　Findings,Hypotheses,Default-Hypotheses,そして Label

- def-hypoは知識ベースの中にはassertされないので、def-hypo Dの環境EはDの逆の仮説を結論付けることを妨げるfindingの集合になるように解釈される。
- def-hypoの計算の方法（図１参照）
  - cが知識ベースにインプットされる。
  - ¬Sが真であると仮定する。

- それ故、¬Rが結論付けられる。
- {p,c} が入力されるとRと¬Rの両方が成り立ってしまう。
- 暗黙の仮定として、もしインプットとして与えられなかったら、その否定が成立すると仮定する（閉世界仮説）。例でいくと a はインプットとして与えられていないから¬aが成り立つと仮定する。

①知識ベースの順序付け

- KB-Reducerでは、ルールは非サイクルネットの構造である必要がある。
- 知識ベースが非サイクルであると仮定すると、次のような再帰定義に従って決定されるレベルにルールを分割する事が出来る。
  - もし r の左辺の全ての条件が満たされていれば　rのレベル＝0
  - rのレベル＝1＋（rが依存するルールのレベルの最大値）

②パーシャルラベルとルールラベル

- 知識ベースが非サイクルであるとすると、level-0,level-1,　というように順にルールを処理することにより、知識ベースをreduceすることができる。
- 全てのルールが処理された後、全てのhypothesesとdefault-hypothesesのラベルが分かる。一つ一つ処理されたルールとして生成された不完全なラベルをパーシャルラベル（Partial label）と呼ぶ。
- パーシャルラベルの更新は次の様に行われる。
  - rule rが現在処理されており、rが仮説Hをassertしたとする。
  - 最初にrの左辺にある文字のラベルの論理結合を持ち、そして結果表示を最小化することによってrの左辺を満足させる最小環境の完全な集合を計算する。生成された最小環境の集合を“rule rのルールラベル”と呼ぶ。
  - 仮説Hの新しいパーシャルラベルはrのルールラベルの論理命題と現在のパーシャルラベルにより決定される。
  - もし上記のように計算が行われたら、ルールのルールラベルはそのルールが知識ベースに追加された時に計算可能になり、計算が他のルールに移るとき、ルールのルールラベルが決して更新されないことを証明できる。

	ﾙｰﾙ	ﾚﾍﾞﾙ	ﾙｰﾙﾗﾍﾞﾙ	ﾊﾟｰｼｬﾙﾗﾍﾞﾙ (After Processing Rule)
r1	p∨q→A	0	p∨q	A p∨q
r2	q∨r→B	0	q∨r	A·p∨q,B q∨r
r3	s&¬q→¬B	0	s&¬q	A·p∨q,B q∨r,¬B s&¬q *
r4.	A→D	1	p∨q	A p∨q,B q∨r,¬B s&¬q, D:p∨q
r5:	B→¬D	1	q∨r	A p∨q,B q∨r,¬B s&¬q, D·p∨q,¬D q∨r **
r6.	D&¬A→T	2	(p&¬p&¬q) ∨(q&¬p&¬q)***	A·p∨q,¬A ¬p&¬q,B.q∨r, ¬B s&¬q,D p∨q,¬D q∨r

*印　s&¬q&r as leading to potential contradiction　B and ¬B
**印　q as leading to potential contradiction　D and ¬D
***印　D&¬A→T as unsatisfiable(redundant) rule

図２　非一貫性ルール及び非充足ルールの検出

③冗長性の検査

- 仮説Hをassertするルールのルールラベルを計算するとする。この時次のようなことを決定することによって冗長性のチェックをする。

  i) rのルールラベルは非一貫的または無効な環境から成っていないか?
  rは決して満足されないので知識ベースは冗長である。

  ［例］図２の r6 はルールラベルが(p&-p&-q)v(q&-p&-q)なので絶対に発火しないため冗長なルールであるといえる。

  ii) rのルールラベルは仮説Hの現在のパーシャルラベルに包含されていないか?
  満足させる全てのケースは前に処理されたルールによってすでにカバーされているので、ルールrはKBから削除される。

  ［例］ r11 a ∨ b -> H　　Hのﾊﾟｰｼｬﾙﾗﾍﾞﾙ　(a ∨ b)
  r22 a -> H　　r22のﾙｰﾙﾗﾍﾞﾙ　(a)
  r22のルールラベルは現在のHのパーシャルラベルに包含されているため、r22は冗長ルールである。

  iii) rのルールラベルは仮説Hの現在のパーシャルラベルを包含していないか?
  一つまたはそれ以上、前に処理されたルールによって結論付けた全てのケースに於けるHをルールrは結論付けるので、前に行われたルールのいくつかが知識ベースから削除される。
  これは、上記例の反対のケースである。

④矛盾性の検出

- 仮説Hをassertしたルールrのルールラベルを計算するとする。rのルールラベルを使ってHのパーシャルラベルを更新する。
- Hと競合する全ての仮説Xに対して次の様なsubset/superset テストとunionテストが行われる。
- subset/superset-testはHのパーンャルラベルの中にどんな環境もあるかを決定する。このHのパーシャルラベルはXのパーシャルラヘルの中にあるとんな環境のsubsetまたはsupersetである。
- この様な環境はXとHの両方を知識ベースにassertする入力集合を表現する。環境はフラグが付けられる。

  ［例］　図２の r5 においてDと¬Dにのパーンャルラベルを見る。
  D：ｐ∨ｑ
  ¬D：ｑ∨ｒ
  ｐ∨ｑの環境とｑ∨ｒの環境は相反する仮説を導く。

- E1とE2が、どちらも他のsubsetでないようなX，Hのパーンャルラベルから作られる環境としよう。その様な各ペアーにたいしてunion-testは結合されたE1∪E2の環境が意味的制約を侵していないかをテストする。もし、E1∪E2がそれらの制約を侵していないならば、それは、矛盾の仮説を知識ベースにassertさせた入力の組み合わせを表している。

  ［例］　図２において r3 のBと¬Bのパーンャルラヘルのunionをとった環境を見てみる。
  ｓ＆¬ｑ＆ｒの環境を作りだしたとすると、この環境は意味的制約を侵していないがBと¬Bの両方を導く。

（５）おわりに

①実験結果

- このシステムはExplorer -IIで走行している。
- 実験結果を表１に示す。

表１　KB-Reducerの実験結果

ルール数	50	150	370
処理時間	40cpu-s	5cpu-m	10cpu-h
生成環境数	700	4000	35000

- いくつかのケースにおいて冗長なルールを見つけることが出来た。
- 注目すべきことは、１回このシステムを走らすと、知識ベースを修正する影響を決定するので知識ベース全体に対して再処理を施す必要が無くなることである。

②複雑性の考慮事項

- 知識ベースの順序付けの複雑さはルールの数に比例する。
- 縮小化のステップの複雑さは生成される環境の数に比例する。
  最悪3^nの環境を生成する（n：findの数）。

③今後の方向

- 知識ベースのサイクリックを許すこと、推論モデルが選べること、データ指向を弱めること（ある文字が入力にも仮説にも両方使える）等を現在研究している。
- 変数を持ったルールが扱え、述語論理の特徴をもっと生かしたKB-Reducerを開発したい。

12. Digital社におけるコンフィギュレーションエキスパートシステム

タイトル：	Expert Systems for Configuration at Digital: XCON and Beyond
著　者　：	Baker,V E and O'Connor,D E
出　典　：	Communications of the ACM,Vol 32,No.3, pp.298-318,(Mar 1989)
分　類　：	Experience
ｷｰﾜｰﾄﾞ　：	CES,XCON,XSEL,XFL,XCLUSTER,XNET,SIZER

（１）まえがき

Digital Equipment Corporation(DEC)では、ＸＣＯＮ等のＣＥＳ（Configuration Expert System）を約十年にわたって開発してきた。ここでは、その経験を以下の観点からまとめる。

- ＥＳの開発、運用を成功させるには、技術的問題以外にビジネスニーズと組織の問題を考慮する必要がある。
- ＥＳの開発には、従来のソフトウェアエンジニアリング以外に独自の技術が必要である。

（２）Digital社におけるＣＥＳの現状

１）稼働開発中のＣＥＳ

①ＸＣＯＮ

主に以下の機能を提供する（原文で述べられている機能の一部を挙げる）。

- 顧客の注文を技術的、営業的視点から解析し、その妥当性を検証する。
- 必要なハートウェアの組み合せ、及びケーブルを求める。

②ＸＳＥＬ

- 顧客の注文に応じて、システムの部品を対話的に選択する。
- ソフトウェアも含めて、部品の組み合せの妥当性を検証する。
- 計算機室に関し、システムの設置に必要な事項を求める。

③ＸＦＬ

- 計算機室におけるシステムのレイアウトを求める。

④ＸＣＬＵＳＴＥＲ

- 通信網のクラスタの設計を支援する。

⑤ＸＮＥＴ

- ローカルエリアネットワークの設計と、その妥当性の検証に用いる。

⑥ＳＩＺＥＲ

- 仕事及び環境に応じて、必要な計算機資源の量を推定するために用いる。

以上のうち、①～④が稼働中、そして⑤、⑥が開発中である。

２）利用状況

ＣＥＳのユーザ部門及びそこでの利用目的は以下の通り。

①販売部門

顧客の注文の妥当性を検証し、見積りを出すための利用。

②製作部門
注文の実現可能性の検証、注文の分割（分担して製作するため）、組立て、検査プログラムの選択。
③フィールドサービス部門
現場での組立て、設置、既存機器との統合。
④製作部門及びエンジニアリング部門
システムレベルの設計の評価。

３）事業への貢献
顧客の満足、生産性の向上、経費削減の点で貢献した。特に、顧客のニーズに合わせてオーダメイドのシステム構成を提供するというDigital社の戦略を支えている。金額換算で、年に4000万ドル以上の貢献がある。具体的な効果は以下の通りである。
①システムの不備がなくなる。また、システムの構成方法が一貫する（同し機能のシステムが常に同し構成になる）ので、顧客を混乱させることがなくなる。
②情報の一元管理により、製作、フィールドサービス部門の混乱を防止する。その結果、製品の立ち上がり、組織の配置変更を迅速化てきる。
③問題のある注文（実現不可能な注文など）を早期に発見し、手戻りを防止できる。
④分担して製作するサブシステムの間の不整合が防止できるので、全体の整合性を確認する工程（Final Assembly and Test）が省略てきる。

（３）ＣＥＳの開発経験
以下では、事業的問題、技術的問題、組織的問題に関する経験及びそこから得られた教訓について述へる。
１）事業的問題
①技術の適用分野
ＥＳのような新技術に対して戦略的投資を行う場合、事業の根幹に係わるような適用分野を選ふへきである。そうすれば、各部署から充分な支援が得られる。ＣＥＳの場合は、オーダメイトのシステムを提供するという会社の戦略を支えるシステムであった。
②ビジネス部門との協力
ＣＥＳの各ユーザ部門のマネージャからなるConfiguration Systems Steering Committeeを設立し、部門間の調整を考慮した開発の方向付けを得ている。さらに、もっと上位レベルの協力を組織化しつつある。
③開発環境
Digital社では、ＥＳ技術に対する期待が大きく、支援が充分てあった。

２）技術的問題
①技術的挑戦
a．既存言語及びツールのうまい利用
パターンマッチング機能の強さと処理の速さを考慮して、VAX OPS5を利用した。
b．ＥＳ構築のためのソフトウェアエンジニアリング方法論の確立

- ソフトウェアエンジニアリング方法論ＲＩＭＥを開発し、ＸＣＯＮの再開発に利用して有効性を明らかにした。
- ＥＳに適したテスト技術の開発が今後の重要課題である。

②アプリケーション特有の問題に対する挑戦

ａ．分野の変化

コンフィギュレーション分野は、関連する事物の変化が多い。そのためＣＥＳのルールの４０％を毎年書き換えている。具体的には、以下の変化がある。

- 毎年膨大な新製品が出る。またそれに伴い既存製品の組み合せ方が変わる。
- 分野知識及びＥＳ技術の蓄積に伴う知識ベースの改良。
- コンフィギュレーション業務自体の変化。（例えば、ネットワークの普及に伴う業務の変化）

ｂ．システムの拡張の必要性

以下の理由から絶えずＣＥＳを拡張する必要がある。

- ユーザ層の拡大に伴うニーズの変化
- 既存ユーザのニーズの変化
- 他のシステムとの結合に伴うニーズの発生
- ユーザが一旦システムを使い始めると要求がどんどん高くなってくる。（例えば、使い勝手の良さ、利用部署毎に最適化した出力など）

ｃ．システムの巨大さと複雑さ

- 各ＣＥＳのルール数を表１に、ルール数の複雑さを表２に示す。
- データベースの規模を表３に示す。

表１　各CESのﾙｰﾙ数（1988/9時点）

ＣＥＳ	ルール数
ＸＣＯＮ	１０１２９
ＸＳＥＬ	３６２９
ＸＦＬ	１８０８
ＸＣＬＵＳＴＥＲ	２４３
ＸＮＥＴ（ﾌﾟﾛﾄﾀｲﾌﾟ）	１７０７

表２　ルールの複雑さ（平均）

パート	要素数
ＩＦ部の条件数	6
条件中の要素数	5
ｔｈｅｎ部のａｃｔｉｏｎ数	4

表３　データベースとその規模

データベース	規　　模
部品ＤＢ	30000以上の部品 部品データの要素数約４０
容器部品ＤＢ（部品を差し込む容器のＤＢ）	７２５以上の容器部品
部品形状ＤＢ(出力のための図形表現のＤＢ)	１２００以上の図形
質問文テンプレートＤＢ	１２００以上の文型
システム構成ＤＢ（ＣＥＳの解答）	稼働箇所により異なる

- 従来型ソフトウェアは３５０個以上のプログラムで、50,000行以上である。

③開発管理

a．巨大さ・複雑さへの対応

従来のソフトウェア開発と同様に、全体のアーキテクチャを定めた後、各部分を開発するアプローチが必要である。

b．保守性の確保

- Digital社のＡＩトレーニングプログラム及びＣＥＳ開発専用のトレーニングプログラムを用いて、人員養成期間を短縮した。
- ＥＳ開発方法論ＲＩＭＥにより、人員養成期間を短縮すると共に、保守に伴う品質の安定を図った。
- 自前のソール群により、知識ベースの頻繁な保守に対応した。

c．短期的開発と長期的開発のバランスに注意した。

３）組織的問題

開発体制を確立するために、開発に必要な役割(key role)と開発フェースを明確化した。

①ＥＳ開発に必要な役割を表４に示す。

表4　ＥＳ開発に必要な役割

役　　割		仕　　事
CHAMPION		戦略的ビジョン、信念、政治力を持ってSPONSORを動かし開発環境を整備する。
SPONSOR		ＥＳの目的達成に関して責任を持ち、リソース、資金を提供して必要な環境を与える。
PROGRAM MANAGER		プログラム開発を管理する。
PROGRAM OFFICE		PROGRAM MANAGERを補佐。 業務分析や開発計画の立案を行う。
TECHNICAL TEAM	KNOWLEDGE ENGINEER	ＡＩ部分の設計、知識獲得。
	SOFTWARE SYSTEMS INTEGRATION ENGINEER	ＥＳの全体設計、非ＡＩ部分の設計、開発、開発環境の整備、ﾘﾘｰｽ管理
	RESEARCH / ADVANCED DEVELOPER	開発が成功した段階で次の新技術を導入する。
	EXPERTS	対象分野の知識を提供する。
	USERS	対象分野のビジネス環境、そこでのニーズなどに関する知識の提供。

②役割に関する教訓を以下に述べる。

a．ｿﾌﾄｳｪｱｼｽﾃﾑｲﾝﾃｸﾞﾚｰｼｮﾝｴﾝｼﾞﾆｱの役割を過小評価し、人員を充分割当てしなかったため、大きな手戻りが生じた。

b．専門家は複数必要である。一人の専門家に頼ると偏ったＥＳになる。

c．技術的問題に注意が偏り、ユーザの意見を軽視した結果、大きな手戻りが生じた。そこで偏りを防止するために役割間に次のような関係を持たせた。

- プログラムマネージャは、スポンサとユーザに対して責任を持つ。
- テクニカルチームは、プログラムマネージャに対して責任を持つ。
- ユーザは、ＥＳ導入によるビジネス環境の向上に関し、スポンサに対して責任を持つ。

③ＥＳ開発モデルに関しては、以下のタスクからなるモデルを確立した。

- システムの定義及び再定義

ＥＳ技術の利用方法、従来技術との役割分担、ユーザニーズの充足などの

問題

- システムの拡張、洗練化
  知識の獲得と管理、複数システムの統合、各種インタフェースの管理
- システムの配布
  リリース管理、ユーザサポート
- システムの評価
  上記のモデルは、基本的には従来のソフトウェア開発モデルと同様である。

各タスクの遂行は以下の理由から従来より試行錯誤的になる。

- 技術の発展が速い。
- 分析しにくい問題を対象としている。
- 技術と問題の相性がよく分かっていない。
- ES導入によりビジネス環境が大きく変わる。

（4）結　論

CESのDigital社に対する貢献をまとめると以下のようになる。

①複数の部署にまたがるコンフィギュレーション業務及び関連情報を統一的に整理し明確化している。

②計算機製品及びその組み合せが次第に複雑化していく中で、オーダメイトのシステムを作るという会社の戦略を支えている。

③業務を効率化し業務変化への対応を容易化し、業務形態の発展を促進している。

④ES技術の適用実験として、様々な知見をもたらしている。

13. エキスパートシステムのテスト、検証、引渡し －Mentor履歴の事例－

タイトル：	Testing, Verifying, and Releasing an Expert System The Case of History of Mentor)
著　者　：	Cochran,E L and Hutchins,B
出　典　：	Proc 3rd Conference on Artificial Intelligence Application,pp 163-167,(Feb 1987)
分　類　：	Experience
ｷｰﾜｰﾄﾞ　：	Mentor,ﾌｨｰﾙﾄﾞﾃｽﾄ,導入時ﾕｰｻﾞ訓練ﾌﾟﾛｸﾞﾗﾑ,技術移転

（１）概　要

MentorはHoneywell社（Ｈ社）のビルサービス代理店が大規模冷却システムを維持管理する際に、Ｈ社のビルサービス部門（BSD: Building Services Division）が作業支援するのに使用する携帯用エキスパートシステム（ES: Expert System）である。MentorはPML（（Ｈ社に）知的所有権のある「知識工学用手続き言語」）で作成され、ラップトップコンピュータ上で実行される。Mentorの知識ベースは、600 個以上のルール形式風のアクションから成っている。これらのアクションは、サービス対象の冷却システムの型式、サービスを行う原因、サービス対象の冷却システムの過去のサービス履歴、及び特定のシステム構成に応じたシーケンスである。本論文では、徹底的なフィールドテスト、知識ベースの妥当性検査、そして諸支援サービス（ユーザ訓練プログラムなど）の開発などのESの最終引渡しを行うために実施した導入準備作業の過程について述べている。ESに関する技術移転は、いくつかの点でユニークであり、Mentorの引渡し準備作業の間に学んだ教訓は、ESがより日常的なツールとなってきているので、広範囲に適用できると予期している。

（２）フィールドテストの諸計画

１）Mentorの特徴

Mentorは、ESとして見た場合、通常のESと比べ次の諸点で異なっている。

①典型的なユーザは、最初はES及びAIはいうまでもなく、キーボード、ディスク装置、パーソナルコンピュータに馴れていない。

②電圧測定器が道具であるように、それと同じ感覚で使えるように、Mentorを道具として設計した。Mentorは数百人のエンドユーザに使用されることを想定している。エントユーザすべては、その道具を使用して、仕事をより完璧に、より効率よく行うことに関心を持っている。

③Mentorには、米国中どこでも使用でき、異なる型式の装置、異なる気候風土、及び異なる使い方のパターンに課せられた多様な冷却システム維持管理要件を満たすことが期待されている。

④Mentorを実行するハードウェアは、携帯用で、堅牢で（サービス用のトラックのバックに何週間も入れっぱなしにしておいても故障しない）さらに、機械室の天井裏に簡単に持ち運びができ、汚く、雑音が多く、振動の激しい環境でも使用できなければならない。

2）フィールドテストの要件

Mentorのフィールトテストの計画が作成されるにつれて、空調をサーヒスするのに使用する方法(method)は、次に示す1次要因及び2次要因により、非常にバラエティに富んでいることがはっきりしてきた。

①1次要因

- サービス対象の機械の型式（遠心式冷却装置対往復運動式冷却装置）
- 付属装置の型式（バックアップポンプ、浄化装置など）

②2次要因

- 気候風土
- 使用地域の地理上の位置
- ビルサービス代理店の経営資源
- ビルサービス代理店の経験度及び従業員の選り好み

フィールトテストの目的が、Mentorの能力及び弱点を明確にすることならば、次に示す内容を満たす広範なテストが行われなければならない。

①異なる地域に属する地方で
②異なる型式の冷却装置
③異なるエントユーザ
④異なる季節

フィールトテスト計画は、次に示す諸点により複雑になってきた。

①テスト可能期間が短い（4カ月）
②Mentorに対する洞察的な分析能力を備えた高度な経験を持つヒルサーヒス代理店の従業員をたくさん必要とするが、不足している。
③サービスを必要とする多様な冷却システムを扱っているBSDの支局数が比較的少ない。

3）フィールトテスト計画の策定及び実施

Mentor用の十分なフィールトテスト計画を作成することがてきた。フィールトテスト計画では、Mentorのすへての機能に対し、複数のテストサイト及び冷却装置設備を使用して、徹底的なテストを実施することにした。結果として、次に示す条件で、Mentorに対し、冷却システムの正規の使用（年間て典型的な冷却装置に対して15回サービス訪問すると仮定・定義し、換算して）て約6年間の使用累積結果が収集できた。

①4カレンダ月
②東部、北東部、中西部、そして西部にある6カ所のBSDフィールトオフィス

4）フィールトテストの結果

Mentorの診断能力は、フィールトテスト期間中、期待以上てあった。Mentorはサービス対象の装置に発生している多種多様な問題を正しく検出し、かつ、すへてのケースにおいて、ビルサービス代理店の従業者が気付いた問題すへてを検出した。Mentorは、通常、検出した問題に対する正しい手順（解決策）を勧告する（勧告文が不完全あるいは誤っている場合には、KBが修正される）。1つ以上の原因による冷却設備の長期間解決できなかった問題に対しても、Mentorは正しい処置方法（アクション）を勧告した。冷却装置に問題が発生し、実際にサーヒス訪問し、Mentorを使用した場合、Mentorの能力は良好であったが，ヒルサーヒス

代理店の従業員が模擬的に訪問して課した特殊な問題の診断に関しては、KBにいくつかの弱点が現われた。加えて、「ユーザインタフェース」、「導入時ユーザ訓練プログラム」、そして「Mentorシステム起動手順」がフィールトテストの初期の頃、混乱の原因となった。結果として、Mentorの様々な側面が改善された。「システムコート」、「KB」、「ユーザインタフェース」、「導入時ユーザ訓練プログラム」、そして「Mentorシステム起動手順」が，フィールトテストの結果、すべて変更され、変更のいくつかは大規模であった。例えば、KBはフィールトテスト期間中に30%も増大し、ユーザインタフェースは完全に改善された。フィールトテストの結果として、ESのテスト及び妥当性検査がどういう意味をもつかに関して、我々の認識が変化した。発生した論争点は、最後に教訓という形で提示する。

（２）フィールトテスト結果に基づく変更

フィールトテストデータの分析結果に基づく変更は、次に示す３つに類型化された。

①ソフトウェア誤りによる変更
②KBに内在する問題
③システムとユーザとのインタフェース問題

（３）技術移転の過程で学んた教訓

Mentorは商用として広範囲に使用する目的て引渡されている数少ないESの１つであり、技術移転の過程で我々は貴重な経験をし、ESを成功に導く手順について学んた。以下に、重要な教訓を示す。

①十分な計画が要諦である。

技術移転ては、関心やニーズの異なるたくさんのグループの人々を巻き込むので、これらの人々の関心やニーズを満たす詳細な計画か必要てある。

②ターゲットグループの人達を開発の初期の段階から参画させ、そして開発の最初から参画しているグループの人達を継続的に参画してもらうような計画を立てよ。技術的に高度な製品を、ターゲットグループに対し、あたかも既成の事実のようにして提供した場合には、技術導入は成功しそうにもない。仮に、最初はユーザに受け入れられたとしても、その開発に参画してないグループによる支援ではうまくゆくはずがない。支援に関する技術的要件、システムの複雑さに馴れるには長期間必要、そして、NIH（not invented here,他人の発明したものは使わない）症候群は、成功をさまたげる強固な障壁てある。

③KBの検証は複数の専門家にやってもうらこと。

この教訓は知識工学界で論争の種になっているが、我々の経験ては、ESを知識のある広範なエントユーザに受け入れてもらうには、誰もが受け入れることのできるアプローチ（取り組み方）を提供するか、あるいは、異なる環境ごとに適する様々なアプローチを提供するかのいずれかてなければならない。いずれの方法を採用するにしても、１人の専門家のみに頼るKEは、代替アプローチに対し、システム開発期間中及びユーザ訓練期間中に、準最適として、システムに組み入れるのか、あるいは明示的に棄却するのかの決定をやらないという危険を冒すことになる。

④フィールドテストは早期に、頻繁に行うこと。

ESのテストを研究所内に限定することもできないし、プロジェクトの開発者のみで実施することもできない。代わりに、テストは、実際の使用環境下で、実際に使用するユーザをできるかぎり十分に反映したサンプルで行われなければならない。テストの目的は、KBの問題点を同定するだけでなく、不十分なユーザインタフェース、不十分なユーザ訓練なと、ユーザ側の受け入れに際しての障壁をも同定することであり、この点が肝心である。

今後とも、Mentorの進化過程の監視を続ける予定である。ESが現場てより長期間使用されることにより、今後の技術移転やES支援活動を計画するに際して貴重な更なる経験を積むことができるであろう。

14. ORにおける視覚対話型モデリング手法：その成功と可能性

タイトル： Visual Interactive Modelling in Operational Research. Successes and Opportunities
著　者　： Bell,P C
出　典　： Journal of the Operational Research Society, Vol.36,No 11,pp 975-982, (1985)
分　類　： Others
ｷｰﾜｰﾄﾞ　： ｺﾝﾋﾟｭｰﾀｸﾞﾗﾌｨｯｸｱﾌﾟﾘｹｰｼｮﾝ、視覚対話型モデリング

（１）概　要

視覚対話型モデリング手法は今やＯＲ専門家にとって一般的なツールの一部であり、それに関する新しいアプリケーションが定期的に報告されるようになっている。本論文では視覚対話型モデリング実施の状況を考察し、成功アプリケーション領域および将来の可能性を特定する。

（２）はじめに

視覚対話型モデリング手法（Visual Interactive Modelling.ＶＩＭ）は単なる一手法ではなく幅広い対話型のグラフィックなモデル構築方法に対して用いられる一般的な用語である。ＶＩＭによるモデルには二つの基本的なタイプがあり、グラフによる表現モデル（Representational Graphic Models ＲＧＭ）とアイコンによるグラフィックモデル（Iconic Graphic Models ＩＧＭ）である。

ＲＧＭは表現のためにチャートや図形（ヒストグラム、線グラフ、パイチャート等）を用いて数学的なモデルからのデータをサマリーする。ＲＧＭは主要なコンピュータグラフィックスの商業的アプリケーションであるし、今後もそうであろう。

しかし、このタイプのモデルは概念的には数学的モデルそのものからほとんど進んでいない。したがって問題は表形式の表現に対するグラフィックな表現の価値、および色による表現の価値があるかどうかである。これらの問題の結果は明確である。例えばＩｖｅｓはグラフィックスの使用に関して管理の生産性を向上したとか、より高度の決定をもたらしたとかいう決定的な証拠はないとまで言っている。

ＩＧＭは問題あるいはプロセスをグラフィック表示としてモデリングする。特に興味をそそられる分野は対話環境におけるＩＧＭである。最も成功したＩＧＭのアプリケーション分野はＰＥＲＴ／ＣＰＭなどのプロジェクト管理の分野である。さらにＶＩＳ（Simulation）アプリケーションには、待行列解析、ジョブショップスケジューリング、交通量解析、組立てライン設計、コンベイヤシステム設計、生産計画、多目的計画等がある。地図に基づくＶＩＭは、経路問題、輸送計画、施設配置問題等に良く用いられている。トラベリングセールスマン問題に対する視覚対話型アプローチはHurrionにより試みられており、100都市から200都市に適用された時には、最適値の４％以内の平均精度の答えを与えるとしている。

ＶＩＭは問題解決のためのツールであるばかりではなく、注意を換気しＯＲ専門家の仕事を表現する理解容易な手段を提供する。

（３）現在のＶＩＭの問題点

現在のＶＩＭの中心的な問題は今後のユーザにとってあまりにも難しく、使い勝手が悪いことである。ＶＩＭシステムを構築するには複雑て込み入ったコンピュータプログラミングを必要とする。ＶＩＭの使い勝手はモデルヒルダとエントユーザという二つのユーザがいることによりより複雑になっている。モデルビルダは通常専門家であり、エントユーザは通常意志決定者（管理者またはプラントオペレータ）、または教育または訓練を受けている学生てある。ＶＩＭがエントユーザの要求により良く答えられるようにするためにエントユーザとモデルビルダを区別することは有効である。

現在の技術はモデルビルダが一人の人間ではなくＯＲ専門家とコンピュータプログラマから成り立つチームであることを要求する。ＯＲ専門家は視覚モデルを設計し、そのモデルがスマートに見えるように数学的モデルを構築するための適当な問題解決技法を選択しなければならない。プログラマはコーディングによりグラフィックスおよび対話インターフェィス、さらにはモデルやアルゴリズムを開発しなければならない。

エンドユーザにとって現在の技術に欠けているものは、モデルの対話要素を理解しモデルを管理者に説明してくれる“お抱えの運転手”がいないことてある。仮にいたとしても、管理者の意思決定は部下の監視にさらされることになり、モデルの実験手段としての有効性は減しられるかもしれない。この“お抱え運転手”の代替案としてはエンドユーザにモデルの使い方を教えることてある。しかしそれは時間的な制約等により上級の管理者による使用を制限することになる。もう一つの代替案はよりつ使い勝手の良いモデルを構築することてある。

ＶＩのエントユーザを助ける上て最も困難なことは、ユーサはとのようにモデルを使用すへきかには注意を払っているにしても、ユーザがとのようにモデルを使用したいかについてほとんと注意が払われていないことてある。モデルはＯＲの専門家の問題解決アプローチに従って問題を考察するように構築され、管理者が彼等のアプローチを使って問題を考察できるようにはなっていない。Parkerは彼のＶＩモデルを構築するのにまず最初幾つかの表示方法を設計し、問題を持っている人とともに検討している。デスカッションと修正の後てこれらのスクリーンはプログラムされる。

管理者が興味を持つ機能はチェックポイント（すなわち、ある決定を試みて、それがうまくゆかないことが判明したら、同し決定ポイントに戻って別の方法を試みることができる機能）、前後への素早い移動画面、タッチスクリーン（部品を自分の指で移動したい）、その他の使い勝手の良い入出力装置（ライトペン、マウス、その他）である。ＶＩＳにとって有効な機能は同時実行可能な複数のシミュレーションである。同しようなンステムの同時ンミュレーション実行は有効であり、あたえられたシステムの複数の意思決定ルールにおけるシミュレーション結果の同時表示は直接の比較を可能にする。

ＶＩマーケティングモデル、ＶＩ金融計画モデル、ＶＩ戦略計画モデル、ＶＩ人員計画モデル、および複数のソフトな意思決定規範が主要な公共部門ては、大きなＶＩマーケティングのポテンシャルがある。これらの分野ては最初はク

ラフィック表現的なＶＩモデルが使われるだろうが、我々がこれら非空間システムの視覚表現について有効なものを開発するにつれて、ＩＧＭがすぐさまそれに続くであろう。

（４）新しい技術

ＶＩＭの発展がハートウェアの発達に依存していることは疑いがない。ＶＩＳはもっと大きなモデルが可能になるより早いプロセススピート、より実際的な時間スケール、より多くの色、よりきめの細かい易しい使い勝手の良い対話が可能になれば大きく進歩するであろう。より簡便で持ち運へるモデルも望ましい。

現在はＶＩＭはソフトウェアに依存しておりソフトウェアに制約されている。ユーザはすべてのレベルにおける操作において問題と制限を感している。ＶＩＭをより簡単に使用し理解できるためにもっと大きなソフトウェアの発展が必要である。いつかは意志決定者は自分自身でＶＩモデルを開発し実行できるようになるだろうが、現在は夢物語てある。

（５）今後の方法論

ＯＲ社会と管理者達によりＶＩＭを受入れてもらうためには幾つかの方法論的な問題が解決されなければならない。その中でも最も重要な鍵は価値を立証することである。我々はバッチモートのモデルよりもＶＩＭが必要なことを実証しなければならない。ＶＩモデルのコストは明確て算定可能たが、その利点は算定不可能で計測もできない。

ＶＩモデルの数学的モデルと比較した妥当性の容易さについて、特にエントユーザによる妥当性検査によって、多くのことが言われたり書かれたりしているが証拠は逸話的である。データの表形式の表示よりグラフィックス表示より価値があるという証拠はほとんどない。ＯＲ専門家は問題解決に対する視覚対話型アプローチの価値について独自の方法で証明しなければならないたろう。

その他の発展が必要な領域は表示そのものの設計である。色や動きは表示の意味を偏向するかも知れない疑いがある。しかし我々がとのように中立的な表示が設計できて意志決定者が彼等の決定が真実彼等自身によるものてあると確信できるという訳てはない。

ParkerとBellはソフトンステム(Soft System)に基ついた視覚対話型問題解決（Visual Interactive Problem Solving ＶＩＰＳ）の方法を提唱している。創造性研究（ブレインライティング(brain writing)、カイデットファンタジィ(guided fantasy)）からのアイデアを利用することにより、この創造的な分野は発展の大きな機会がある。

（６）結　論

ＶＩＭは実に短かい期間に多くの成功をおさめ、ＯＲの重要な部分になってきた。管理者に考えられているようにＶＩＭの出現はＯＲにとって良い影響をあたえている。次の４ステップがこの良い立場を強固なものにするであろう。

最初に視覚対話型手法はモデル構築手法というよりも問題解決アプローチとして強調されるへきてある。すなわち、ＶＩＭよりもＶＩＰＳが強調されるへきである。次に、ＶＩＭは単なる視覚対話型シミュレーションより以上のもので多くのタイプの問題が利益を受けるものであることが強調されるへきてある。

第３番目に、グラフィックス表示の設計により注意が払われるべきである。最後に、ＶＩＰＳは上級の管理者が一晩中起きていられるような重要な意思決定問題に成功裡に適用されるべきである。

ＯＲ社会はこの機会を捕え、視覚対話型手法が育つにつれて多くの利益を得られるようにすべきである。

付録C. 文献一覧表

[会議名の略称一覧]

AAAI: I-th National Conference on Artificial Intelligence (American Association for Artificial Intelligence)
AISIG: Artificial Intelligence Systems in Government Conference
CAIA: The i-th Conference on Artificial Intelligence Application
ESIG: I-th Annual Expert Systems in Government Conference (Expert Systems in Government Symposium)
IJCAI: The i-th International Joint Conference on Artificial Intelligence
TAI : The i-th International IEEE Workshop on Tools for Artificial Intelligence
WESTEX: Western Conference on Expert Systems

[文献の分類キーワード一覧]

[BASIS] : 方法論的基盤(Methodological Basis)
[VERIF] : 検証(Verification)
[VALID] : 妥当性検査(Validation)
[CR&EV] : 信用性・評価(Credibility and Evaluation)
[MD&TQ] : 手法・技法(Methods and Techniques)
[STATI] : 統計手法(Statistical Tools)
[AUTOM] : 自動化(Automation)
[EXPER] : 事例(Experience)
[LF-CY] : ライフサイクル(Life Cycle Issues)
[OTHER] : その他(Others References)

[1983]

Gasching, J., Klahr, P., Pople, H., Shortliffe, E., and Terry, A., "Evaluation of Expert Systems: Issues and Case Studies," in Building Expert Systems, ed. by Hayes-Roth, F., Waterman, D.A., and Lenat, D.B., MA:Addison-Wesley, Reading, 1983, pp.241-280.
（なぜＥＳの評価をするのか；ＥＳの評価を設計する際の問題点；評価すべきＥＳの特徴；いつＥＳを評価するのか；どのようにＥＳを評価するのか；事例：Ｒ１）
[VALID]

Miller, F.D., Copp, D.H., Vesonder, G.T., and Zielinski, J.E., "The ACE Experiment: Initial Evaluation of an Expert System for Preventive Maintenance," Joint Services Workshop on Artificial Intelligence in Maintenance, Oct. 1983, pp.421-428.
（ＡＣＥ（Ａutomatic Ｃable Ｅxpert）；電話網の保守過程でのＡＣＥの役割；フィールドでのＡＣＥの能力を評価する際に使用した手法；ＥＳ評価に関する一般的な論争点）

【EXPER】
Vesonder, G.T., Stolfo, S.J., and Zielinski, J.E.,
"ACE: An Expert System for Telephone Cable Maintenance,"
IJCAI-83, pp.116-121, (Aug. 1983).
（ＡＣＥ（Ａutomatic Ｃable Ｅxpertise）；電話網保守の際に必要とするトラブルシューティング報告書及び管理資料を作成するＥＳ；構築の詳細；ＡＣＥのテスト；1982年春以降フィールド・テスト中）
【EXPER】

【1984】
Buchanan, B.G. and Shortliffe, E.H., "The Problem of Evaluation,"
in Rule-Based Expert Systems, ed. by Buchanan, B.G. and
Shortliffe, E.H., MA:Addison-Wesley, Reading, 1984, pp.571-588.
【CR&EV】
Kohoutek, H.J., "Quality Issues in New Generation Computing,"
FGCS'84, pp.695-702, (Oct. 1984).
（次世代コンピューティングの品質問題；知識，知識表現，ＫＢ，探索過程，演繹推論，帰納推論，学習，プログラミング言語の品質；ハードウェアの品質）
【BASIS】
Stevens, A., "How Shall We Judge on Expert Systems," in Expert Systems:
Principles and Case Studies, ed. by Forsyth, R., London:Chapman and
Hall Computing, Reading, 1984, pp.36-48.
（ＥＳの社会的影響；ＥＳの評価；薬の臨床試験との対比；医師の診断との対比）
【CR&EV】
Suwa, M., Scott, A.C., and Shortliffe, E.H., "Completeness and
Consistency in a Rule-Based System," in Rule-Based Expert Systems,
ed. by Buchanan, B.G. and Shortliffe, E.H., MA:Addison-Wesley,
Reading, 1984, pp.159-170.
（ルールベースの完備性及び一貫性に関する原著論文）
【VERIF】
Yu, V.L., et al, "An Evaluation of MYCIN's Advice," in Rule-Based
Expert Systems, ed. by Buchanan, B.G. and Shortliffe, E.H.,
MA:Addison-Wesley, Reading, 1984, pp.589-596.
【EXPER】

【1985】
Bell, M.Z., "Why Expert Systems Fail," Journal of the Operational
Research Society, Vol.36, No.7, pp.613-619, (1985).
【CR&EV】
Bell, P.C., "Visual Interactive Modelling in Operational Research:
Successes and Opportunities," Journal of the Operational Research

Society, Vol.36, No.11, pp.975-982, (1985).
【CR&EV】
Kline, P.J. and Dolins, S.B., Choosing Architectures for Expert Systems, Final Technical Report RADC-TR-85-192, Oct. 1985, pp.154, Rome Air Development Center, Griffiss AFB, NY, 13441.
（ＥＳの設計ガイドライン；問題の特性に適したＡＩ技術の選定；53個のＡＩ技術の解説有）
【MD&TQ】
Liebowitz, J., "Evaluation of Expert Systems: An Approach and Case Study," CAIA'85, pp.564-571, (Dec. 1985).
（1986年版 "Useful Approach for Evaluating Expert Systems" の元論文）
【EXPER】
Neches, R., Swartout, W.R., and Moore, J., "Explainable (and Maintainable) Expert Systems," IJCAI-85, pp.382-389, (Aug. 1985).
【OTHER】
Nguyen, T.A., Perkins, W.A., Laffey, T.J., and Pecora, D., "Checking an Expert Systems Knowledge Base for Consistency and Completeness," IJCAI-85, pp.375-378, (Aug. 1985).
（ルールベースの完備性及び一貫性に関する論文）
【VERIF】

【1986】
Adelman, L., "On the Basis of What Criteria should Decision Support Systems be Evaluated?," ESIG'86, pp.348-350, (Oct. 1986).
【MD&TQ】
Chandrasekaran, B., "Generic Tasks in Knowledge-Based Reasoning: High-Level Building Blocks for Expert System Design," IEEE EXPERT, Vol.1, No.3, pp.23-30, (Fall 1986).
【OTHER】
Fieschi, M. and Joubert, M., "Some Reflections on the Evaluation of Expert Systems in Medicine," Methods of Information in Medicine, Vol.25, No.1, pp.15-21, (Jan. 1986).
（医療分野でのＥＳの能力の問題点；入力データの信頼性，ルールの信頼性，確信度計算，説明機能，診断の一致性についての検討）
【EXPER】
Frail, R.P. and Freedman, R.S., "OPGEN Revisited: Some Methodological Observations on the Delivery of Expert Systems," ESIG'86, pp.310-317, (Oct. 1986).
（ＯＰＧＥＮ（Ｏperations Ｓheets Ｇenerator）；開発方法論；ＨＡＩＬ－１，ＯＰＧＥＮ－Ｒ，ＯＰＧＥＮ－Ｌの比較評価；教訓的ガイドライン）
【EXPER】
Freedman, R.S. and Frail, R.P., "OPGEN: The Evolution of an Expert System for Process Planning," AI Magazine, Vol.7, No.5, pp.58-70,

(Winter 1986)
（上記の論文の改訂版）
【EXPER】
Hansen, J V and Messier, W F Jr , "A Knowledge-Based Expert System for Auditing Advanced Computer Systems," European Journal of Operational Research, Vol.26, No 3, pp 371-379, (1986)
（コンピュータ監査のＥＳ；コンピュータ・システムの内部統制の信頼性；コンピュータ監査へのＥＳ適用の可能性；ＡＬ／Ｘ）
【EXPER】
Hansen, J V and Messier, W F Jr , "A Preliminary Investigation of EDP-XPERT," Auditing A Journal of Practice & Theory, Vol 6, No 1, pp.109-123, (Fall 1986)
（ＥＤＰ－ＸＰＥＲＴ（コンピュータ監査用ＥＳ）の評価；ＥＳの評価手法；評価基準；評価結果；ＡＬ／Ｘの評価）
【EXPER】
Kline, P J and Dolins, S B , "Problem Features that Influence the Design of Expert Systems," AAAI-86, pp 956-962, (Aug 1986)
（1985年版 "Choosing Architectures for Expert Systems" の要約論文）
【MD&TQ】
Langlotz, C P , Shortliffe, E H , and Fagan, L M , "Using Decision Theory to Justify Heuristics," AAAI-86, pp 215-219, (1986)
【MD&TQ】
Liebowitz, J., "Useful Approach for Evaluating Expert Systems," Expert Systems, Vol 3, No 2, pp 86-96, (Apr 1986)
（訳「エキスパート・システムをいかに評価するか ある評価手法の実験」，日経ＡＩ，第16号付録，pp 10, (1986 8 4)）
（専門業務をＥＳでやるべきか，専門家がやるべきかの選定手法；ＡＨＰ（Analytic Hierarchy Process）；Read（Requirements Engineering Automated Development））
【EXPER】
Oliver, A.E M , "Techniques for Expert System Testing and Validation," 3rd International Expert Systems Conference, Learned Information, pp.271-276, (1986)
（従来のソフトウェア開発で使用されているテスト技法及び妥当性検査技法をどのようにすればＥＳに適用できるか；ＥＳの妥当性検査の経験）
【MD&TQ】
Richer, M H., "An Evaluation of Expert System Development Tools," Expert Systems, Vol 3, No 3, pp 166-183, (July 1986)
【MD&TQ】
Vera, J.W. and Bloczak, R , "Artificial Intelligence Techniques Applied to Claims Processing Expert System Research Tool," ESIG'86, pp.116-127, (Oct 1986)
【OTHER】

[1987]
Citrenbaum, R., Geissman, J.R., and Schultz, R., "Selecting A Shell," AI EXPERT, Vol.2, No.9, pp.30-39, (Sept. 1987).
[MD&TQ]
Cochran, E.L. and Hutchins, B.L., "Testing, Verifying, and Releasing an Expert System: The Case History of Mentor," CAIA'87, pp.163-167, (Feb. 1987).
（Mentor（ビルの空調システムの予防保守ＥＳ）；フィールド・テスト；ＫＢの妥当性検査；ＥＳ引渡しの準備作業；教訓）
[EXPER]
Cragun, B.J. and Steudel, H.J., "A Decision-Table-Based Processor for Checking Completeness and Consistency in Rule-Based Expert Systems," Int. J. Man-Machine Studies, Vol.26, No.5, pp.633-648, (1987).
（ルールベースの完備性及び一貫性に関する論文）
[AUTOM]
Davis, G.A. and Yeh, D., "An Expert System for Selecting Appropriate Domains for Expert Systems," ESIG'87, pp.260-265, (Oct. 1987).
[MD&TQ]
Green, C.J.R. and Keyes, M.M., "Verification and Validation of Expert Systems," WESTEX-87, pp.38-43, (June 1987).
（本格的なＥＳのＶ＆Ｖの論文；ＥＳのＶ＆Ｖの悪循環；ＥＳのＶ＆Ｖ方法論（要件の定義，検証，テスト・ケースの準備，テストの実施，評価）の提案）
[VERIF]
Herbert, M.R. and Williams, G.H., "An Initial Evaluation of the Detection and Diagnosis of Power Plant Faults Using a Deep Knowledge Representation of Physical Behaviour," Expert Systems, Vol.4, No.2, pp.90-99, (May 1987).
[EXPER]
Johnson, R.G., Joly, G.C., and King, P.J.H., Survey of Techniques for Checking of Rule-Based Expert Systems, Technical Report, Department of Computer Science, Birkbeck College, London University, 1987, pp.8.
（ルールベースの完備性及び一貫性に関するサーベイ論文）
[MD&TQ]
Johnson, R.G., Joly, G.C., and King, P.J.H., BIRCH - The BIrkbeck Rule CHecker - An Example, Technical Report, Department of Computer Science, Birkbeck College, London University, 1987, pp.8.
（ルールベースの完備性及び一貫性に関する論文）
[AUTOM]
Kjeldsen, R. and Cohen, P., "The Evolution and Performance of the GRANT System," CAIA'87, pp.272-276, (Feb. 1987).
[EXPER]
Lundsgaarde, H.P., "Evaluating Medical Expert Systems," Social Sience and Medicine, Vol.24, No.10, pp.805-819, (1987).

（医療ＥＳ評価の文献調査；医療ＥＳの内の約90％は臨床環境下て評価されていない；医療ＥＳの評価の基本的な手法；研究戦略）
【EXPER】
Marcot, B , "Testing Your Knowledge Base," AI EXPERT, Vol 2, No 8, pp 42-47, (July 1987)
【VERIF】
Marcus, S , Stout, J , and McDermott, J , "VT An Expert Elevator Designer that Uses Knowledge-Based Backtracking," AI Magazine, Vol 8, No 4, pp 41-58, (Winter 1987)
（ＶＴ（Ｖertical Ｔransportation）；エレベータ設計支援ＥＳ；知識獲得ツールＳＡＬＴを使用して生成された）
【OTHER】
Messier, W F Jr and Hansen, J V , "Expert Systems in Auditing The State of the Art," Auditing A Journal of Practice & Theory, Vol 7, No 1, pp 94-105, (Fall 1987)
【EXPER】
Mettrey, W , "An Assessment of Tools for Building Large Knowledge-Based Systems," AI Magazine, Vol 8, No 4, pp 81-89, (Winter 1987)
【MD&TQ】
Nguyen, T A , Perkins, W A , Laffey, T J , and Pecora, D , "Knowledge Base Verification," AI Magazine, Vol 8, No 2, pp 69-75, (Summer 1987)
【VERIF】
O'Keefe, R M , Balci, O , and Smith, E P , "Validating Expert System Performance," IEEE EXPERT, Vol 2, No 4, pp 81-90, (Winter 1987)
（検証・妥当性検査・評価とは；ＥＳの妥当性検査の考慮事項；基本的な諸概念；定性的妥当性検査；定量的妥当性検査；統計的考え方の適用）
【VALID】
O'Leary, D E , "Validation of Expert Systems - with Applications to Auditing and Accounting Expert Systems," Decision Sciences, Vol 18, No 3, pp 468-486, (Summer 1987)
【VALID】
Pau, L F , "Prototyping, Validation and Maintenance of Knowledge Based Systems Software," ESIG'87, pp 248-253, (Oct 1987)
【LF-CY】
Soloway, E , Bachant, J , and Jensen, K , "Assessing Maintainability of XCON-in-RIME Coping with the Problems of a VERY Large Rule-Base," AAAI-87, pp 824-829, (July 1987)
（ＸＣＯＮ（eXpert ＣＯＮfigurer）；ＸＣＯＮの保守面の問題；ルール用高位言語ＲＩＭＥ；コートの同質性・予測性；プロクラミング環境；所見）
【LF-CY】

[1988]
荻野，市川，「エキスパート・システムの運用／評価に関する一考察」，昭和63年度人工知能学会全国大会(第２回)，pp.393-396，(1988.7.21-23).
[LF-CY]
Bahill, A.T., Harris, P.N., and Senn, E., "Lessons Learned Building Expert Systems," AI EXPERT, Vol.3, No.9, pp.36-45, (Sept. 1988).
[EXPER]
Bielawski, L. and Lewand, R., "Testing, Validation, and Maintenance," in Expert Systems Development: Building PC-Based Applications, MA:QED Information Sciences, Reading, 1988, pp.221-239.
[LF-CY]
Burns, J.R. and Morgeson, J.D., "A Methodology for Formulation of Knowledge-Based Simulation Models," Information and Decision Technologies, Vol.14, No.1, pp.15-30, (1988).
[MD&TQ]
Cohen, P.R. and Howe, A.E., "How Evaluation Guides AI Research," AI Magazine, Vol.9, No.4, pp.35-43, (Winter 1988).
[OTHER]
Cupello, J.M. and Mishelevich, D.J., "Managing Prototype Knowledge/Expert System Projects," C.ACM, Vol.31, No.5, pp.534-541, (May 1988).
[LF-CY]
Francioni, J.M. and Kandel, A., "A Software Engineering Tool for Expert System Design," IEEE EXPERT, Vol.3, No.1, pp.33-41, (Spring 1988).
[OTHER]
Geissman, J. R. and Schultz, R. D., "Verification & Validation of Expert Systems," AI EXPERT, Vol.3, No.2, pp.26-33, (Feb. 1988).
[VALID]
Ginsberg, A., "Knowledge-Base Reduction: A New Approach to Checking Knowledge Bases for Inconsistency & Redundancy," AAAI-88, pp.585-589, (1988).
[AUTOM]
Gordon, S., "The Human Factor in Expert Systems," AI EXPERT, Vol.3, No.1, pp.55-59, (Jan. 1988).
[OTHER]
Jacob, R.J.K. and Froscher, J.N., "Facilitating Change in Rule-Based Systems," in Expert Systems: The User Interface, ed. by Hendler, J.A., New Jersey:Ablex, Reading, 1988, pp.249-284.
[LF-CY]
Kirk, D.B. and Murray, A.E.(SAIC), Verification and Validation of Expert Systems for Nuclear Power Plant Applications: Methodology and Life-Cycle, EPRI Report NP-5978s, Aug. 1988.
[VERIF]
Lindsay, S., "Identifying Problems that Expert Systems Can Solve," in Practical Applications of Expert Systems, MA:QED Information

Sciences, Reading, 1988, pp 39-48
（ＥＳを適用すへきアプリケーションを選定するためのチェックリスト）
【MD&TQ】
O'Leary, D E , "Methods of Validating Expert Systems," Interfaces, Vol 18, No 6, pp.72-79, (Nov -Dec 1988)
【MD&TQ】
Parsaye, K , "Acquiring & Verifying Knowledge Automatically," AI EXPERT, Vol 3, No 5, pp 48-63, (May 1988)
【AUTOM】
Parsaye, K and Chignell, M , "Measuring Expert System Performance," in Expert Systems for Experts, New York John Wiley & Sons, Reading, 1988, pp 365-374
（分類型ＥＳの妥当性検査；知識用物差し；混同マトリックス；混同コスト・マトリックス；可能混同マトリックス；リスクのコスト換算）
【VALID】
Slagle, J R and Wick, M R., "A Method of Evaluating Candidate Expert System Applications," AI Magazine, Vol 9, No 4, pp 44-53, (Winter 1988)
（ＥＳを適用すへきアプリケーションの選定手法；ユーザ及びマメジメント，タスク，専門家の各側面から点数付けをし，荷重平均を計算して選定；チェックリストとして有用）
【MD&TQ】
Walters, J and Nielsen, N R , "Evaluating the Prototypes," in Crafting Knowledge-Based Systems, New York John Wiley & Sons, Reading, 1988, pp 121-143
（ＥＳプロトタイプの評価；テストの目標（完備性，一貫性，強固性（robustness））；テスト技法；デバッキング；妥当性検査）
【LF-CY】

[1989]
竹内啓他（編），統計学辞典，東洋経済新報社，1989年12月，pp.1185.
（テストの妥当性，一致係数，信頼区間，同時信頼区間，分割表などの統計関連用語が参考になる）
[STATI]
Bachant, J. and Soloway, E., "The Engineering of XCON," C.ACM, Vol.32, No.3, pp.311-317, (Mar. 1989).
（ＸＣＯＮの問題点（複雑性，修正容易性）；問題解決手法；ルールベース作成のＲＩＭＥ版ガイドライン；ＲＩＭＥの利点；構成用大規模ＥＳ（２～３万ルール））
[LF-CY]
Balci, O., "How to Assess the Acceptability and Credibility of Simulation Results," Proc. of the 1989 Winter Simulation Conference (WSC'89), pp.62-71, (Dec. 1989).
[MD&TQ]
Banks, J., "Testing, Understanding, and Validating Complex Simulation Models," Proc. of the 1989 Winter Simulation Conference(WSC'89), pp.549-551, (Dec. 1989).
[MD&TQ]
Barker, V.E. and O'Connor, D.E., "Expert Systems for Configuration at Digital: XCON and Beyond," C.ACM, Vol.32, No.3, pp.298-318, (Mar. 1989).
[EXPER]
Becker, L.A., Green, P.E., Duckworth, R.J., Bhatnagar, J., and Pease, A., "Evidence Flow Graphs for VV&T," Preliminary Proceedings of the Workshop on Verification, Validation and Testing of Knowledge-Based Systems at IJCAI-89, pp.15, (Aug. 1989).
[MD&TQ]
Bellman, K.L., Combs, J.B., and Stachowitz, R.A.(eds.), Preliminary Proceedings of the Workshop on Verification, Validation and Testing of Knowledge-Based Systems at IJCAI-89, Detroit, Michigan, (Aug. 19, 1989).
[VERIF]
Benbasat, I. and Dhaliwal, J.S., "A Framework for the Validation of Knowledge Acquisition," Knowledge Acquisition, Vol.1, No.2, pp.215-233, (June 1989).
[VALID]
Carson, J.S., "Verification and Validation: A Consultant's Perspective," Proc. of the 1989 Winter Simulation Conference(WSC'89), pp.552-558, (Dec. 1989).
[MD&TQ]
Chang, C.L., Stachowitz, R.A., and Combs, J.B., "Testing Integrated Knowledge-Based Systems," TAI-89, pp.12-18, (Oct. 1989).
[AUTOM]

Constantine, M.M. and Ulvila, J.W., "Knowledge-Based Systems in the Army : The State of the Practice and Lessons Learned, with Implications for Testing," Preliminary Proceedings of the Workshop on Verification, Validation and Testing of Knowledge-Based Systems at IJCAI-89, pp.15, (Aug. 1989).
[EXPER]

Culbert, C. and Savely, R.T., "CRSV and AAMP: Ongoing Work at NASA/ Johnson Space Center in KBS Verification and Validation," Preliminary Proceedings of the Workshop on Verification, Validation and Testing of Knowledge-Based Systems at IJCAI-89, pp.9, (Aug. 1989).
[AUTOM]

Gaines, B.R. and Shaw, M.L.G., "Comparing the Conceptual Systems of Experts," IJCAI-89, pp.633-638, (Aug. 1989).
[BASIS]

Guida, G., "Assuring Adequacy of Expert Systems in Critical Application Domains: A Constructive Approach," in The Reliability of Expert Systems, ed. by Hollnagel, E., Chirchester:Ellis Horwood, Reading, 1989, pp.134-167.
[VERIF]

Harrison, P.R., "Testing and Evaluation of Knowledge-Based Systems," in Structuring Expert Systems: Domain, Design, and Development, ed. by Liebowitz, J. and De Salvo, D. A., New Jersey:Prentice Hall, Reading, 1989, pp.303-329.
[VALID]

Hayes-Roth, F., "Towards Benchmarks for Knowledge Systems and Their Implications for Data Engineering," IEEE Transactions on Knowledge and Data Engineering, Vol.1, No.1, pp.101-110, (Mar. 1989).
(訳:「知識システムのより良いベンチマーク基準に向けて」, 日経ＡＩ別冊 1990冬号, pp.104-117, (1990.1.20).)
[MD&TQ]

Keyes, J., "Why Expert Systems Fail," AI EXPERT, Vol.4, No.11, pp.50-53, (Nov. 1989).
[CR&EV]

Landaurer, C., "Principles of RuleBase Correctness," Preliminary Proceedings of the Workshop on Verification, Validation and Testing of Knowledge-Based Systems at IJCAI-89, pp.21, (Aug. 1989).
[VERIF]

Laswell, L.K., "Testing," in Collision: Theory vs. Reality in Expert Systems, MA:QED Information Sciences, Reading, 1989, pp.205-219.
[LF-CY]

Luqi, "Rapid Prototyping Languages and Expert Systems," IEEE EXPERT, Vol.4, No.2, pp.2-5, (Summer 1989).
[LY-CY]

Marcus, S. and McDermott J., "SALT: A Knowledge Acquisition Language for Propose-and-Revise Systems," Artificial Intelligence, Vol.39, No.1, pp.1-37, (May 1989).
（ルール方式の計画・設計型ＥＳを生成する知識獲得ツール；
Ｐｒｏｐｏｓｅ＆Ｒｅｖｉｓｅ法；ＶＴ（Ｖｅｒｔｉｃａｌ Ｔｒａｎｓｐｏｒｔａｔｉｏｎ）；実用品）
【OTHER】

McGraw, K.L. and Harbison-Briggs, K., "Evaluating Knowledge-Base Development Efforts," in Knowledge Acquisition: Principles and Guidelines, New Jersey: Prentice Hall, Reading, 1989, pp.299-325.
【LF-CY】

Miller, L.A., "Validation and Verification of Knowledge-Based Systems," Invited Tutorial at the Coference on Expert Systems Application for the Electric Power Industry, pp.30, (June 1989).
【VERIF】

Miller, L.A., "A Comprehensive Approach to the Verification and Validation of Knowledge-Based Systems," Preliminary Proceedings of the Workshop on Verification, Validation and Testing of Knowledge-Based Systems at IJCAI-89, pp.31, (Aug. 1989).
【VERIF】

Nazareth, D.L., "Issues in the Verification of Knowledge in Rule-Based Systems," International Journal of Man-Machine Studies, Vol.30, pp.255-271, (1989). Also available in The Foundations of Knowledge Acquisition, ed. by Boose, J.H. and Gaines, B.R., London: Academic Press, Reading, 1990, pp.161-177.
【VERIF】

O'Leary, D.E. and O'Keefe, R.M., Verifing and Validating Expert Systems, IJCAI-89 Tutorial Program MP4, Aug. 21, 1989, pp.37.
【VALID】

O'Keefe, R.M. and O'Leary, D.E., The Verificaton and Validation of Expert Systems, pp.31, 1989.
（上記のチュートリアル資料の改訂版）
【VALID】

Popolizio, J.J. and Cappelli, W.S., "New Shells for Old Iron," Datamation, Vol.35, No.8, pp.41-48, (Apr. 15, 1989).
（メインフレームで使用可能なシェル；ＥＳとトランザクション処理／データベースとの統合；ＡＤＳ（Ａion Ｄevelopment Ｓystem））
【OTHER】

Press L., "Expert System Benchmarks," IEEE EXPERT, Vol.4, No.1, pp.37-44, (Spring 1989).
【MD&TQ】

Ramsey, C.L. and Basili V.R., "An Evaluation of Expert Systems for Software Engineering Management," IEEE Transactions on Software Engineering, Vol.15, No.6, pp.747-759, (June 1989).
（ＡＲＲＯＷＳＭＩＴＨ－Ｐ（ソフトウェア開発プロジェクトの進行状態を診断するＥＳ）；２つの知識獲得アプローチ（ボトム・アップ，トップ・

ダウン）と２つの知識表現（ルール，フレーム）との組み合せた４つのＥＳの比較評価）
【EXPER】
Rossomando, P.J., "A Knowledge Based Environment for the Verification and Validation of Knowledge Based Systems," Preliminary Proceedings of the Workshop on Verification, Validation and Testing of Knowledge-Based Systems at IJCAI-89, pp.15, (Aug. 1989).
【AUTOM】
Szatkowski, G.P., "Approaches to Verification/Validation of Imbedded Decision Supprot Systems Applied to Lunch Vehicle Operations," Preliminary Proceedings of the Workshop on Verification, Validation and Testing of Knowledge-Based Systems at IJCAI-89, pp.8, (Aug. 1989).
【EXPER】
Sizemore, N.L., "Test Techniques for Knowledge Based Systems: A Survey," Preliminary Proceedings of the Workshop on Verification, Validation and Testing of Knowledge-Based Systems at IJCAI-89, pp.16, (Aug. 1989).
【AUTOM】
Weitzel, J.R. and Kerschberg, L., "Developing Knowledge-Based Systems: Reorganizing the System Development Life Cycle," C.ACM, Vol.32, No.4, pp.482-488, (Apr. 1989).
【LF-CY】
Whitner, R.B. and Balci, O., "Guidelines for Selecting Simulation Model Verification Techniques," Proc. of the 1989 Winter Simulation Conference(WSC'89), pp.559-568, (Dec. 1989).
【MD&TQ】
Yoon, J.P., "Techniques for Data and Rule Validation in Knowledge Based Systems," Proc. of the Fourth Annual Conference on Computer Assurance: Systems Integrity, Software Safety and Process Security (COMPASS'89), pp.62-70, (June 1989).
【VERIF】
Zhang, D. and Nguyen, D., "A Technique for Knowledge Base Verification," TAI-89, pp.399-406, (Oct. 1989).
【VERIF】

[1990]
寺野,「知識ベースシステム開発ツールにおけるベンチマーク」, 情報処理, Vol.31, No.3, pp.352-360, (Mar. 1990).
[MD&TQ]
Beckman, T.J., "Methods for Selecting Promising Expert System Applications," AISIG'90, pp.14-21, (May 1990).
[MD&TQ]
Chang, C.L., Combs, J.B., and Stachowitz, R.A., "A Report on the Expert Systems Validation Associoate(EVA)," Expert Systems With Applications, Vol.1, No.3, pp.217-230, (1990).
[AUTOM]
Chapnick, P., "Software Quality Assurance," AI EXPERT, Vol.5, No.2, pp.5-6, (Feb. 1990).
[MD&TQ]
Constantine, M.M. and Ulvila, J.W., "Testing Knowledge-Based Systems: The State of the Practice and Suggestions for Improvement," Expert Systems With Applications, Vol.1, No.3, pp.237-248, (1990).
[EXPER]
Dhaliwal, J.S. and Benbasat, I., "A Framework for the Comparative Evaluation of Knowledge Acquisition Tools and Techniques," Knowledge Acquisition, Vol.2, No.2, pp.145-166, (June 1990).
[OTHER]
Fox, M.S., "AI and Expert System Myths, Legends, and Facts," IEEE EXPERT, Vol.5, No.1, pp.8-20, (Feb. 1990).
(訳:「ＡＩとエキスパート・システム -- その神話と伝説と事実」, 日経ＡＩ別冊1990夏号, pp.134-149, (1990.7.1).)
[OTHER]
Goldstein, D.K., Cho, N., and Zack, M.H., "Knowledge as a Measure of the Effectiveness of Management Support Systems," Proc. of the Twenty-Third Annual Hawaii International Conference on System Sciences, Vol.IV, Jan. 1990, pp.447-457.
[BASIS]
Hajek, B.K., Jakubowski, T., Miller, D.W., and Haggerty, S., "An Off-Line Time Dependent V&V Testing Methodology for a Real-Time Operator Advisor Expert System," Proc. of the Topical Meeting on Advances in Human Factors Research on Man/Computer Interactions: Nuclear and Beyond, pp.40-46, (June 1990).
[MD&TQ]
Hayes, M.M. and Ruoff, K.L., "Benchmark Programs for Evaluating Knowledge-Based Performance Requirements," AISIG'90, pp.6-13, (May 1990).
[MD&TQ]
Hicks, R.C., "A Composite Methodology for Low Maintenance Expert Systems Development," Proc. of the Twenty-Third Annual Hawaii International Conference on System Sciences, Vol.III, Jan. 1990, pp.293-302.

[LF-CY]
Irgon, A., Zolnowski, J., Murray, K.J., and Gersho, M., "Expert System Development: A Retrospective View of Five Systems," IEEE EXPERT, Vol.5, No.3, pp.25-40, (June 1990).
[EXPER]
Jacob, R.J.K. and Froscher, J.N., "A Software Engineering Methodology for Rule-Based Systems," IEEE Transactions on Knowledge and Data Engineering, Vol.2, No.2, pp.173-189, (June 1990).
[OTHER]
Jamieson, R., "Auditing Knowledge-Based Systems," Accounting Research Forum, Sponsored by the Center for Accounting Research, Jan. 24, 1990, pp.36.
[VALID]
Kang, Y. and Bahill, A.T., "A Tool for Detecting Expert-System Errors," AI EXPERT, Vol.5, No.2, pp.46-51, (Feb. 1990).
[AUTOM]
Levinson, S.H. and Daughtrey, H.T, "Issues in Verifying Nuclear Power Plant Expert Systems," Proc. of the Topical Meeting on Advances in Human Factors Research on Man/Computer Interactions: Nuclear and Beyond, pp.33-39, (June 1990).
[VERIF]
Liebowitz, J., "Issues in Verification and Validation of Expert Systems," Proc. of the International Symposium on Knowledge Engineering, Barcelona, Spain, May 1990, pp.8.
[VALID]
Mellis, W., "TWAICE: A Knowledge Engineering Tool," Information Systems, Vol.15, No.1, pp.137-150, (1990).
[AUTOM]
O'Leary, D.E., "On the Representation and the Impact of Reliability on Expert System Weights," in The Foundations of Knowledge Acquisition, ed. by Boose, J.H. and Gaines, B.R., London:Academic Press, Reading, 1990, pp.179-188.
[MD&TQ]
O'Leary, D.E., "Expert System Security," IEEE EXPERT, Vol.5, No.3, pp.59-69, (June 1990).
[OTHER]
O'Leary, D.E. and Watkins, P.R., "A Portfolio of Knowledge Acquistion Approaches for a Knowledge-Based System," Proc. of the Twenty-Third Annual Hawaii International Conference on System Sciences, Vol.III, Jan. 1990, pp.264-268.
[OTHER]
O'Leary, T.J., Goul, M., Moffitt, K.E., and Radwan, A.E., "Validating Expert Systems," IEEE EXPERT, Vol.5, No.3, pp.51-58, (June 1990).
[VALID]

Prerau, D.S., Gunderson, A.S., Reinke, R.E., and Adler, M.R., "Maintainability Techniques in Developing Large Expert Systems," IEEE EXPERT, Vol.5, No.3, pp.71-80, (June 1990).
[LF-CY]

Shaw, M.L.G. and Woodward, J.B., "Validation in a Knowledge Support System: Construing and Consistency with Multiple Experts," in The Foundations of Knowledge Acquisition, ed. by Boose, J.H. and Gaines, B.R., London:Academic Press, Reading, 1990, pp.39-60.
[MD&TQ]

Slagle, J.R. and Gardiner, D.A., "Knowledge Specification of an Expert System," IEEE EXPERT, Vol.5, No.4, pp.29-38, (Aug. 1990).
[OTHER]

Suen, C.Y., Grogono, P.D., and Shinghal, R., "Verifying, Validating, and Measuring the Performance of Expert Systems," Expert Systems With Applications, Vol.1, No.2, pp.93-102, (1990).
[VALID]

Tsai, W.T. and Zualkernan, I.A, "Towards a Unified Framework for Testing Expert Systems," Proc. of the Second International Conference on Software Engineering and Knowledge Engineering, pp.127-134, (June 1990).
[MD&TQ]

Vignollet, L. and Ayel, M, "A Model for Testing Knowledge Bases," Proc. of the Second International Conference on Software Engineering and Knowledge Engineering, pp.104-109, (June 1990).
[MD&TQ]

Wright, R.G., Expert System Verification and Validation, IEEE Managing Expert System Programs and Projects Conference Tutorial Program #2, Sept. 10, 1990, pp.116.
[VALID]

禁無断転載

平成3年3月発行

発行所　財団法人　日本情報処理開発協会
東京都港区芝公園3丁目5番8号
機械振興会館内
TEL03（3432）9372

印刷所　株式会社　タケミ印刷
東京都千代田区神田司町2－16
TEL03（3254）5840

02－R 001